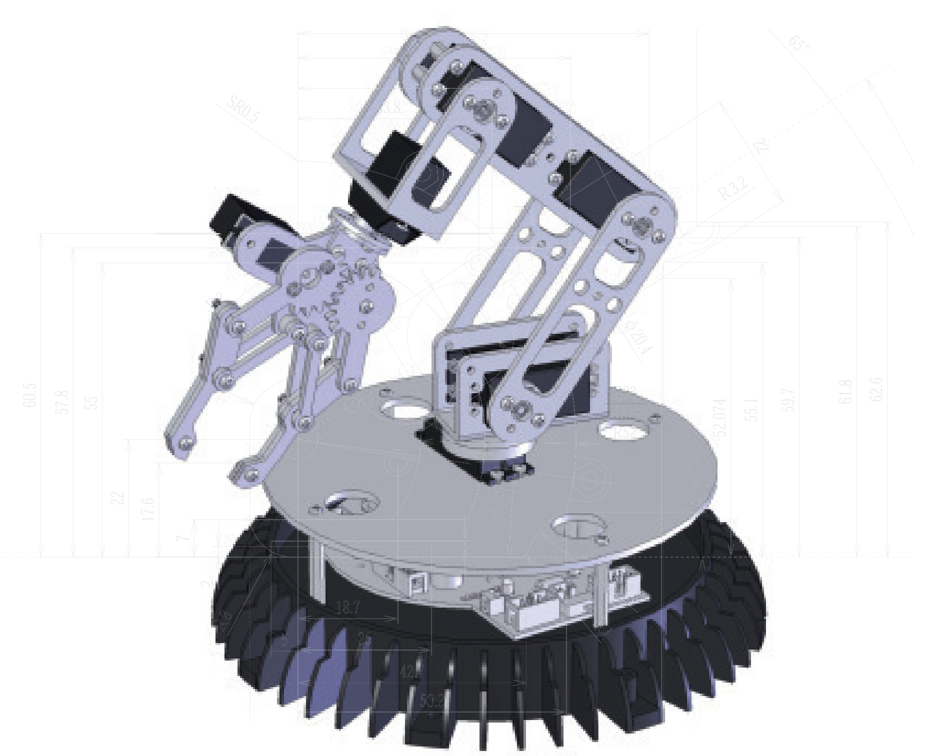




LERNROBOTER
ROBOT ARM PRO

BAUANLEITUNG: Modell RA1-PRO



Inhaltsverzeichnis



Fabrikant:
AREXX Engineering
DAGU HI-TECH



Europäischer Importeur:
AREXX Engineering
ZWOLLE Die Niederlande

Technische Unterstützung beim Bauen des Roboters:

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

1. PRODUKTBESCHREIBUNG ROBOT ARM

Dieser große metallene Roboterarm ist ideal geeignet für Schul- und Lehrprojekte um in die Themen Elektronik, Mechanik und Programmierung einzusteigen. Der Roboterarm wird von einem leistungsstarken ATMEGA64 Mikrocontroller gesteuert, der mit Open Source Tools in C programmiert werden kann. Eigene Programme können einfach über das mitgelieferte USB Interface und Uploader Software in den Mikrocontroller geladen werden. Die I/O Ein-/Ausgänge und das flexible I2C Bus System ermöglichen es, den Roboterarm mit Erweiterungsmodulen auszubauen, damit er auf seine Umgebung reagieren kann.

Lieferumfang:

- *Kompletter Roboterarm-Bausatz (Mechanik und Elektronik)*
- *USB Interface mit Kabel*
- *CD-ROM mit aller notwendigen Software und Anleitungen*

1.2. Spezifikationen:

- ATMEGA64 Prozessor
- Verschiedene freie I/O Ein-/Ausgänge
- I2C Bus
- 6 Maxi-Servos
- Kunststoffarm und Metallgrundgestell
- Armlänge: 390 mm
- Höhe: 460 mm
- Durchmesser Grundgestell: 210 mm
- Stromversorgung: 6-12V

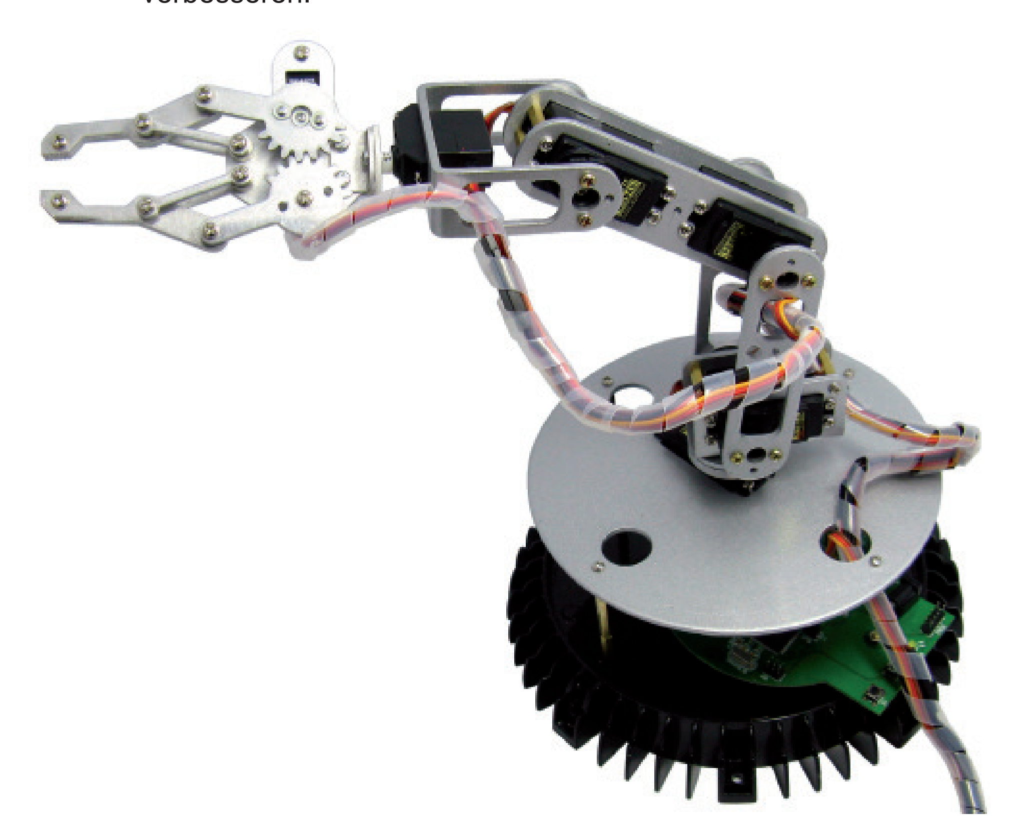


Warnungen

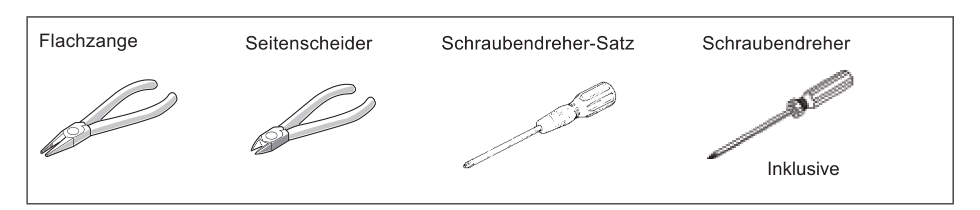
* Mit dem Öffnen der Plastikbeutel mit Komponenten und Teilen erlischt das Rückgaberecht.
* Lesen Sie vor dem Bauen zuerst die Gebrauchsanleitung aufmerksam durch.
* Seien Sie beim Hantieren mit den Werkzeugen vorsichtig.
* Bauen Sie nicht im Beisein kleiner Kinder. Die Kinder können sich an den Werkzeugen verletzen oder kleine Komponenten und Teile in den Mund stecken.
* Achten Sie auf die Polung der Batterien.
* Sorgen Sie dafür, daß die Batterien und die Batteriehalter trocken bleiben. Falls der ROBOT ARM naß wird, entfernen Sie die Batterien und trocknen Sie alle Teile, so gut es geht.
* Entfernen Sie die Batterien, wenn der ROBOT ARM mehr als eine Woche nicht benutzt wird..

1.3. Was können wir mit dem Roboter Arm anfangen?

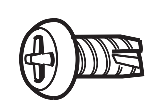
- Beispiel- und Neuprogramme in den Robot Arm übertragen.
- Den Robot Arm mit ein Keyboard kontrolieren
- Den Robot Arm mit der RACS Software steuern und Programieren.
- Den Robot Arm mit gebrauchsfertigen Erweiterungsmodulen ausbauen, so dass er hören, fühlen und sehen kann und somit auf seine Umgebung reagiert
- Wie echte Roboterarme in der Wirklichkeit z.B. Autos bauen, kann auch dieser Roboter Arbeiten für SIe erledigen.
- Über die I2C Schnittstelle kann der Robot Arm mit seinem Umfeld und vielen anderen Geräten kommunizieren.
- Künstliche Intelligenz: mit Hilfe von selbstlernender Software kann der Robot Arm seine Software ständig selbsttätig verbesseren.

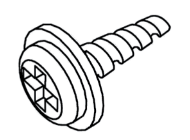


2. Notwendige Werkzeuge

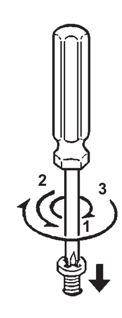


Selbstzapfende Schrauben (Parker)





Schrauben mit einem selbstzapfenden Gewinde verhalten sich wie Holzschrauben, d.h. in einer Drehbewegung schneidet sich die Schraube ein Gewinde und dreht sich dabei fest in das Material. Dazu hat diese Schraubenart ein größeres Gewinde und eine schärfere Spitze als die normale Schraube.

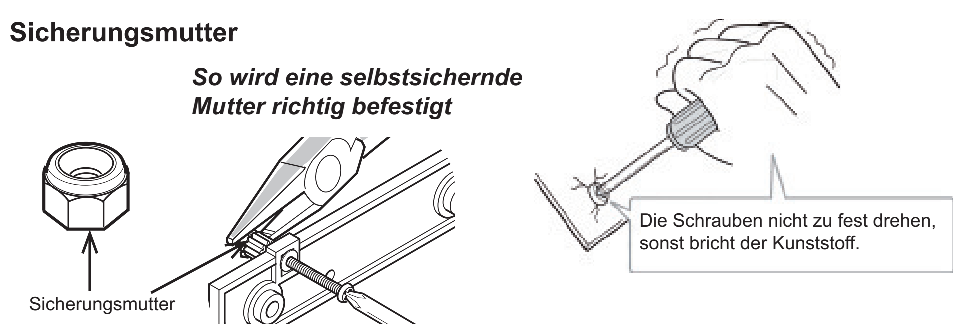


Die Schrauben mit einem selbstschneidenden Gewinde haben an der Spitze auch eine Aussparung, die den Schneidevorgang unterstützt. Die optimale Methode zum Festschrauben einer solchen Schraube ist:

1 Eindrehen der Shraube
2 Leichte Lockerung der Schraube
3 Anschließend wieder Festdrehen

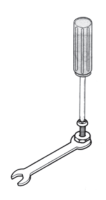
Falls die Schrauben zu oft gelockert und wieder festgeschraubt werden, weitet sich das Schraubloch immer mehr aus und die Schraube paßt nicht mehr richtig.

Sicherungsmutter

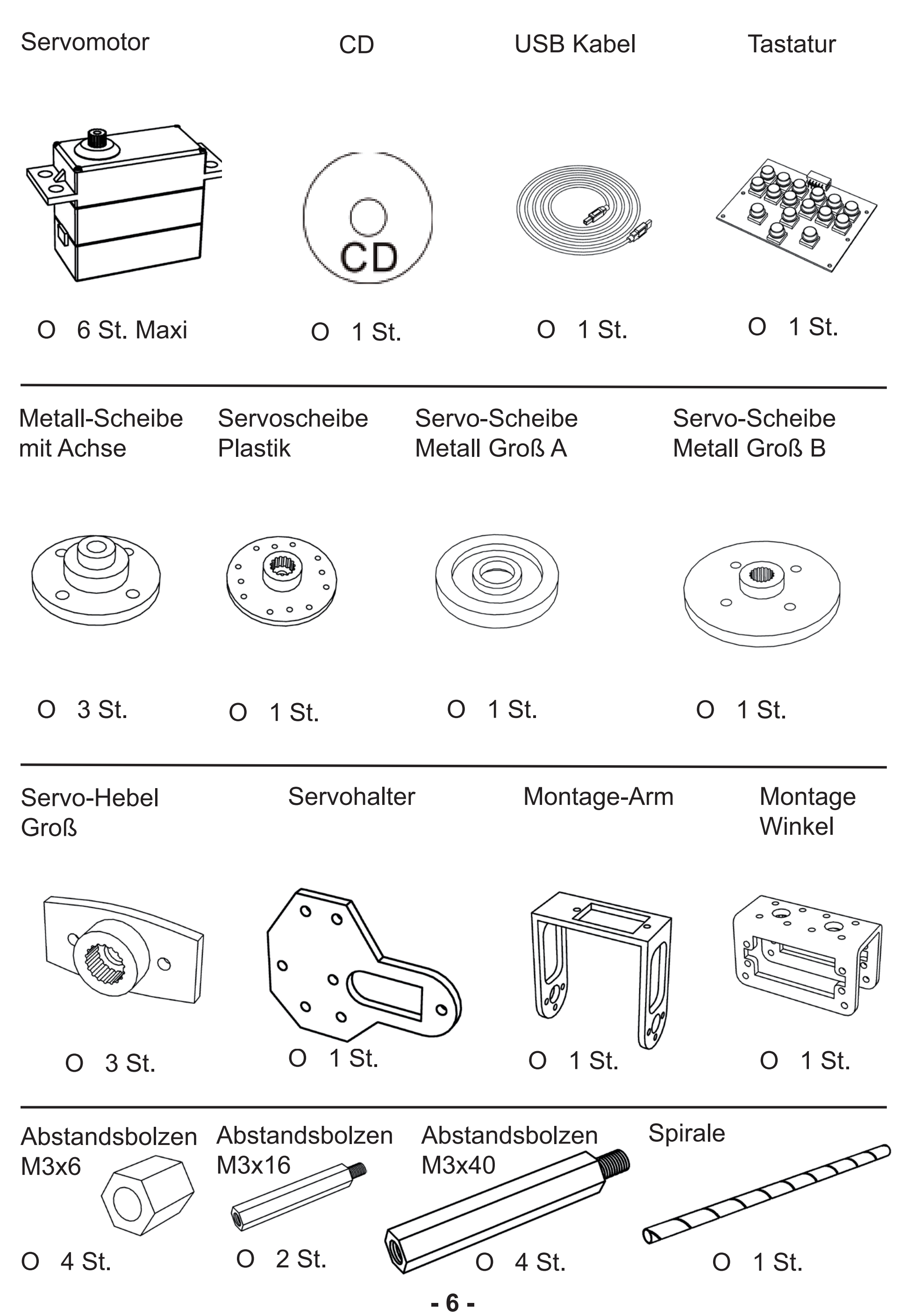


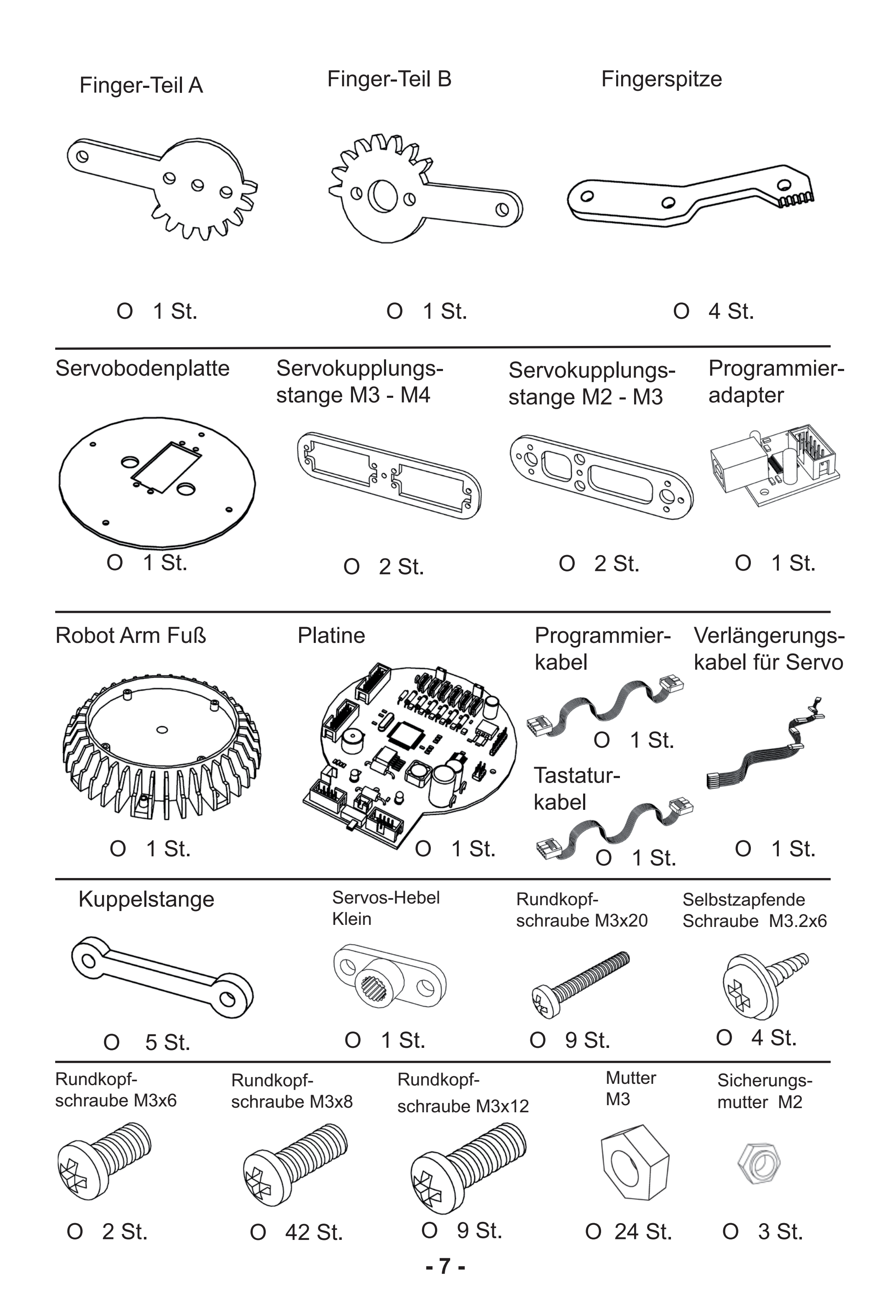
Doppelmaulschlüssel:

Dem Bausatz ist ein Doppelmaulschlüssel beigelegt. Benutzen Sie diesen Schlüssel für die M2 und M3 Mutter. Sie können diesen Schlüssel anstatt einer Zange benutzen.



3. TEILELISTE

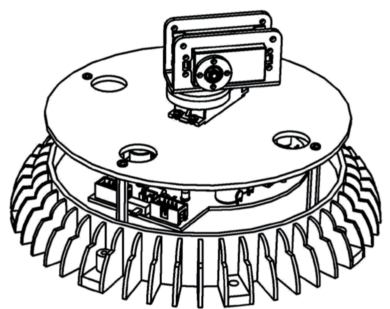



Finger-Teil A
Finger-Teil B
Fingerspitze
O 1 St.
O 1 St.
O 4 St.
Servobodenplatte
Servokupplungs-
stange M3 - M4
Servokupplungs-
stange M2 - M3
Programmier-
adapter
O 1 St.
O 2 St.
O 2 St.
O 1 St.
Robot Arm Fuß
Platine
Programmier-
kabel
Verlängerungs-
kabel für Servo
O 1 St.
Tastatur-
kabel
O 1 St.
O 1 St.
O 1 St.
O 1 St.
Kuppelstange
Servos-Hebel
Klein
Rundkopf-
schraube M3x20
Selbstzapfende
Schraube M3.2x6
O 5 St.
O 1 St.
O 9 St.
O 4 St.
Rundkopf-
schraube M3x6
Rundkopf-
schraube M3x8
Rundkopf-
schraube M3x12
Mutter
M3
Sicherungs-
mutter M2
O 2 St.
O 42 St.
O 9 St.
O 24 St.
O 3 St.

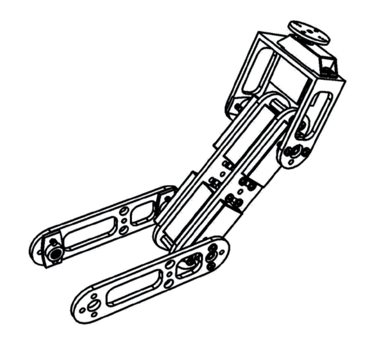
4. Bauanleitung für die Mechanikteile

4.1. Montage Teile in diese Bausatz

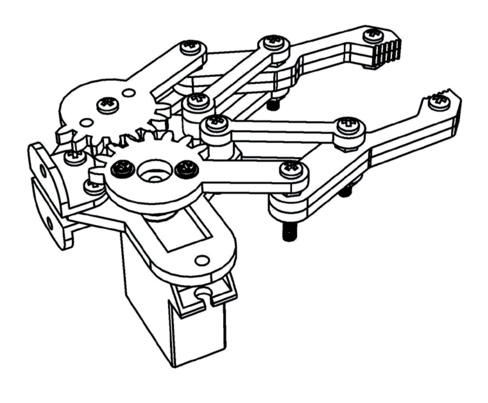
Bodenteil



Arm

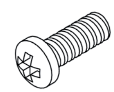


Grijfer



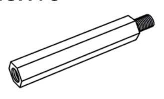
Option
Abstandsbolzen als Kabelführung von Servokabels

Rundkopfschraube
M3x8



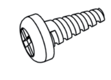
O 8 St.

Abstandsbolzen
M3x16



O 2 St.

Selbstzapfende
Schraube M3.2x8

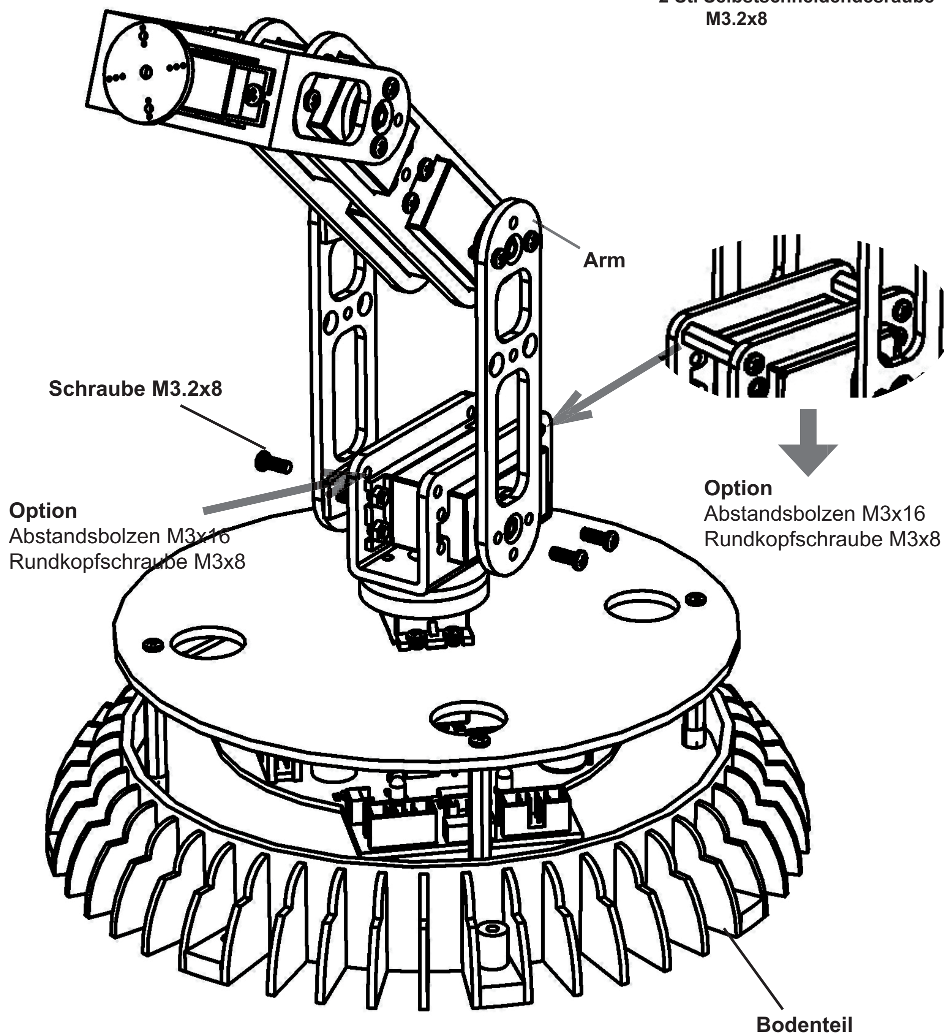


O 2 St.

4.2. Montage des Servo-Arm:

Zur Montage des Servo-Arm wird folgendes benötigt;

1 St. Bodenplatte
1 St. Arm Teil
2 St. Abstandsbolzen M3x16
4 st. Rundkopfschraube M3x8
2 St. Selbstschneidendesraube M3.2x8



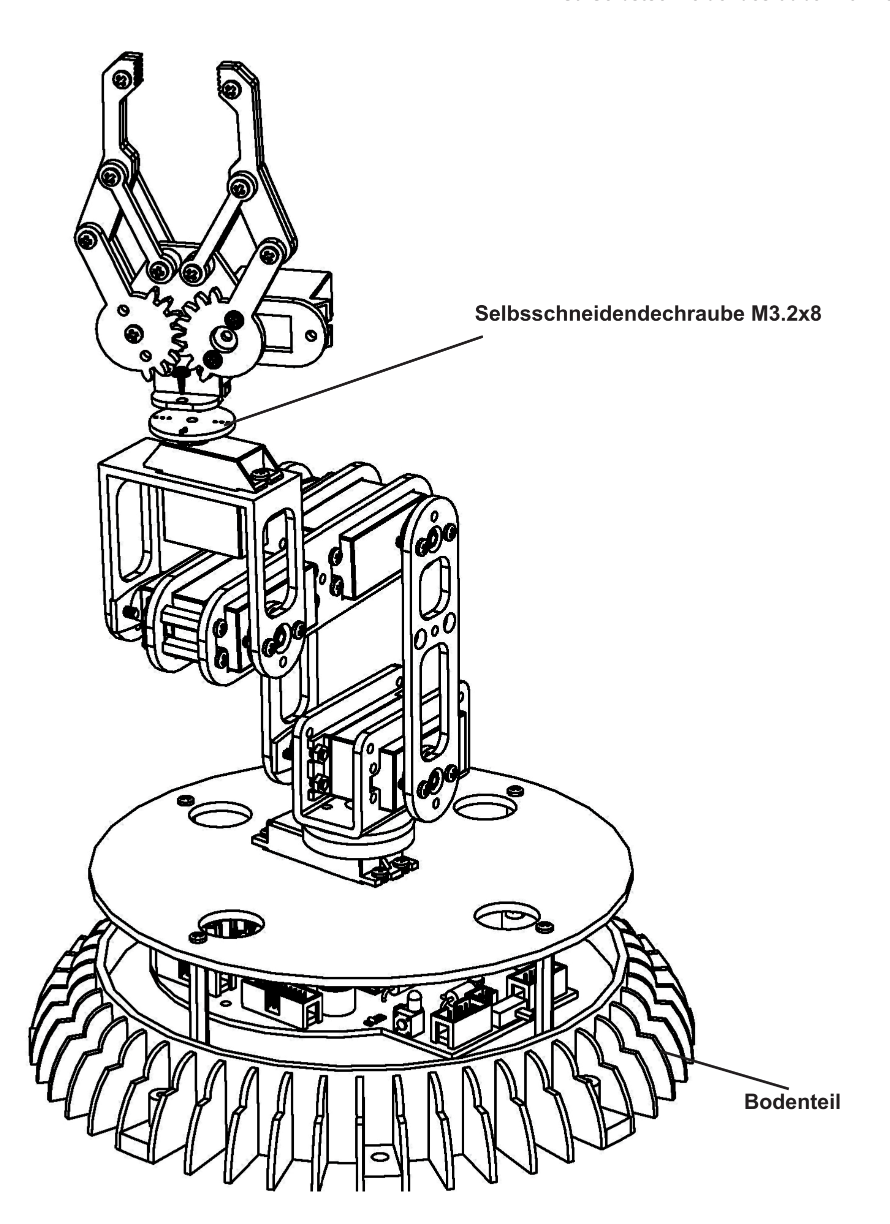
ACHTUNG!

Nehmen Sie erst die komplette Verkabelung vor, bevor Sie diese Schritte ausführen. Den Schaltplan für den Kabelanschluß finden Sie auf Seite 20.

4.3. Montage des Greifer:

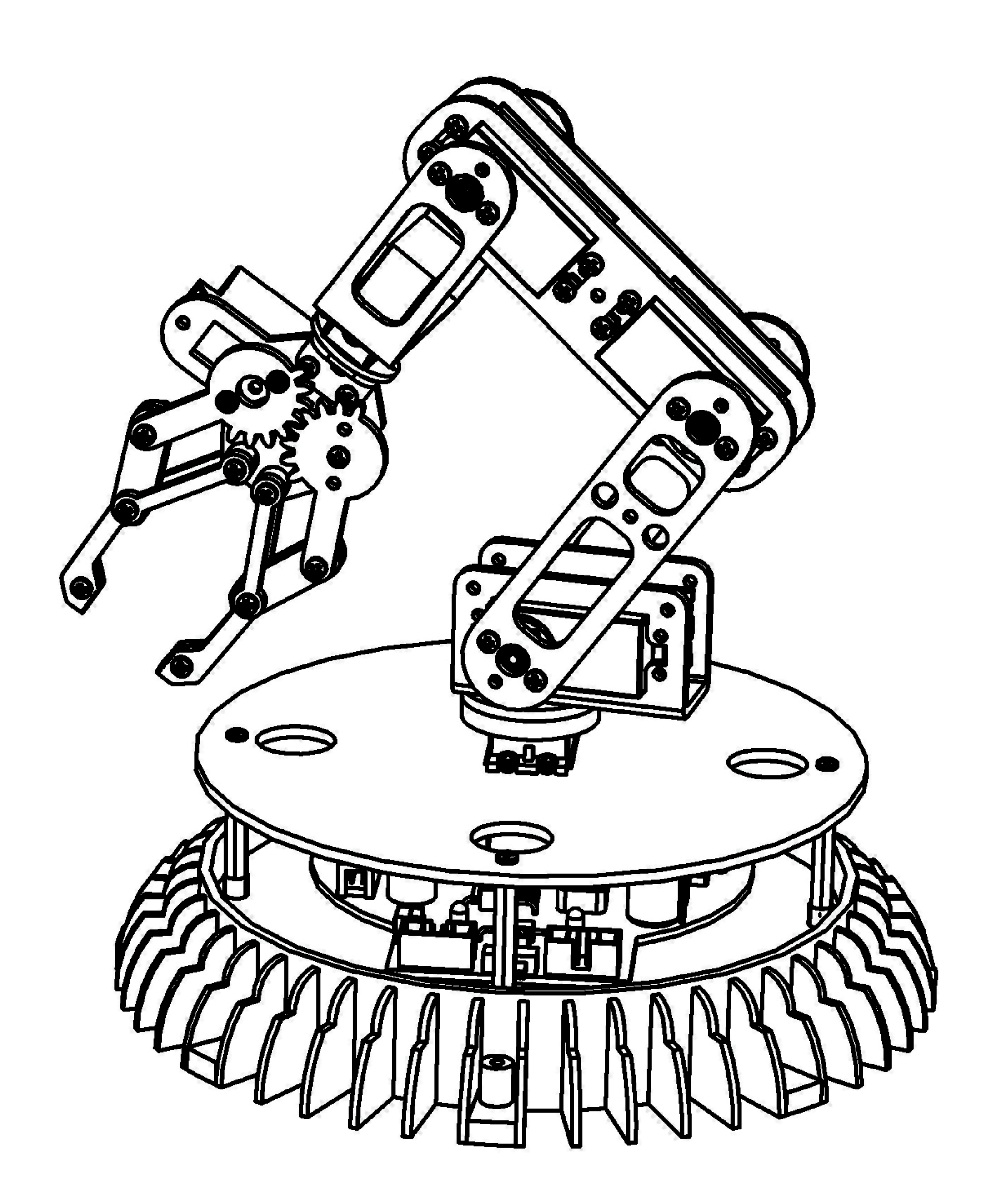
Zur Montage des Grijfer wird folgendes benötigt;

1 St. Boden und Armteil
1 St. Greifer
2 St. Selbstschneidendesraube M3.2x8



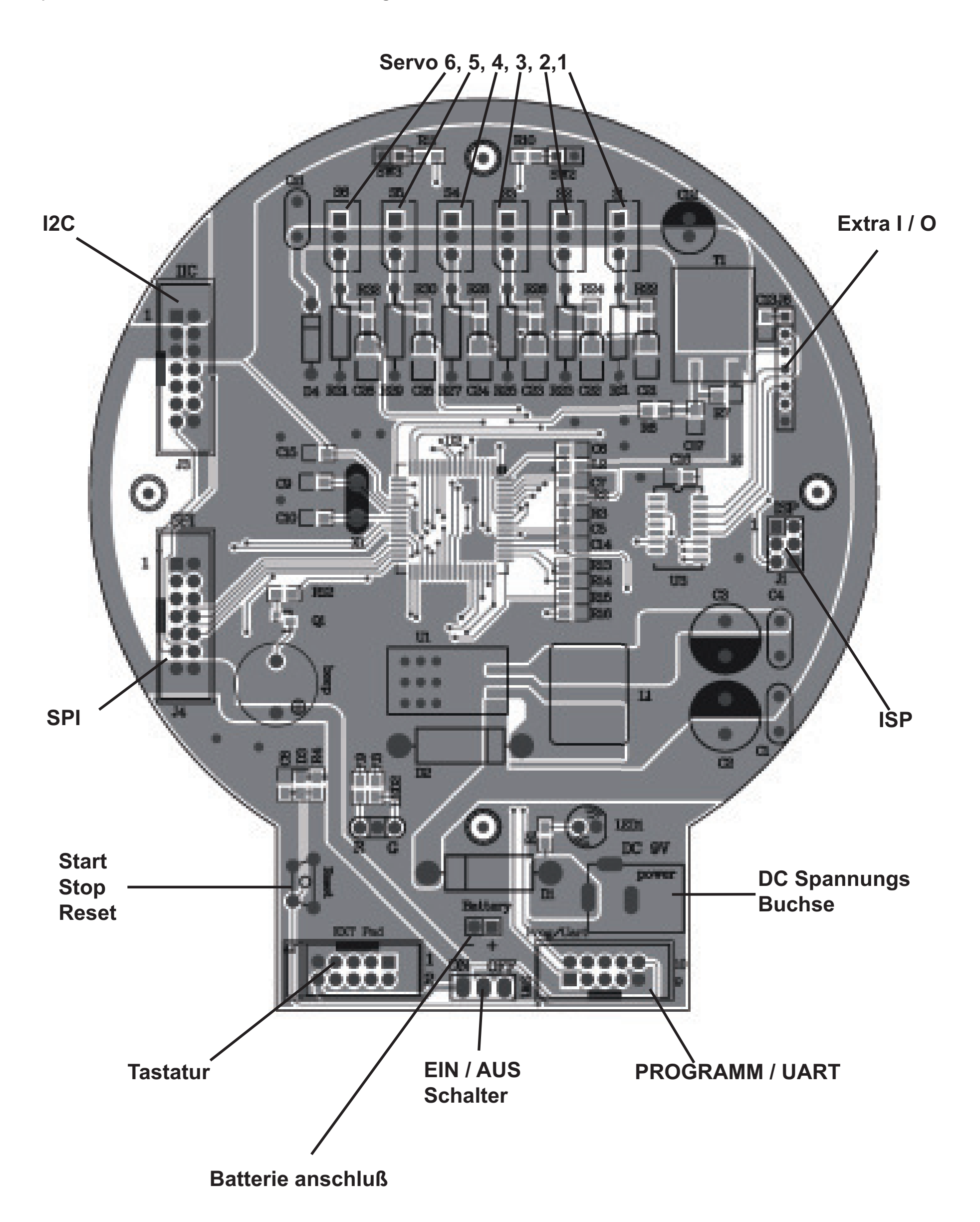
Montieren Sie die Servoachse auf dem Servo, beachten Sie dabei bitte die kleine Detailskizze!

FERTIG !



Anschlussbelegung der Hauptplatine

Schließen Sie die Servos an mit Hilfe von die Servo Verlängerungskabels und benutzen Sie die Spirale, um die Kabel sauber zu verlegen.



5. Die Roboter Arm wird zum Leben erweckt

1. Setzen Sie zuerst die Mechanik- und Elektronikmodule des Roboter Arms mit Hilfe der Bauanleitung zusammen.
2. Schließen Sie ggf. das 9 Volt Netzgerät an (7- bis maximal 12 Volt).
3. Schalten Sie bitte den Roboter Arm mit dem Hauptschalter ein.

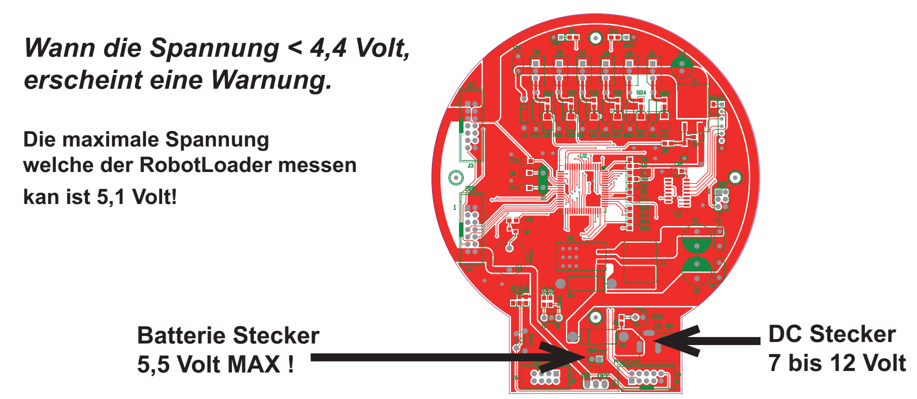
Spannung anschließen

Netzgerät
Es gibt 2 Möglichkeiten zur Stromversorgung. Die einfachste Lösung ist der Anschluss eines Netzgeräts mit einer Spannung von 7 bis 12 Volt / 2 Ampere auf den DC Stecker DC 9V Power. Dabei wird die Spannung auf den Eingang des Spannungsreglers angeschlossen.

Batterien
Die zweite Möglichkeit ist die Verbindung einer Batterie mit dem Batterieanschluss***. Dabei wird die Batteriespannung nach dem Ausgang des Spannungsreglers angeschlossen und darf deshalb auch niemals größer werden als 5,5 Volt!!*** **Falls Sie 4 Stücke normale 1,5 Volt Mono Batterien (D Zellen) verwenden, sollten Sie (in Durchlass-Richtung) eine Diode in Serie mit der positiven Anschlussleitung schalten.** Noch besser geeignet sind vier Stück der großen 1,2 Volt Mono Akkus D.

Wann die Spannung < 4,4 Volt, erscheint eine Warnung.

Die maximale Spannung welche der RobotLoader messen kan ist 5,1 Volt!



Sobald der Roboter Arm auf einer Stromversorgung angeschlossen wird, bewegen sich die Servos oft wenig und leuchtet die Gelbe LED (LED1) auf.

Nun, das war für den Anfang gar nicht so schwierig und es sieht aus, als ob man jetzt bereits fertig wäre. Jetzt aber fängt die Arbeit erst richtig an ...... !

Aber ..... zuerst kommt Kapitel 6, in dem wir die Software installieren

6. Software Installation

Als nächstes kommen wir zur Software Installation. Die korrekt installierte Software wird für alle nachfolgenden Kapitel unbedingt benötigt.

Es sind Administrator-Rechte erforderlich, also melden Sie sich ggf. vorher als Administrator in Ihrem System an!

Wir empfehlen Ihnen zuerst das gesamte Kapitel in Ruhe durchzulesen und erst dann Schritt für Schritt die Installationsanleitung durchzugehen!

Grundlegende Kenntnis der Bedienung von Computern mit Windows oder Linux Betriebssystemen und den gängigen Programmen wie Dateimanager, Webbrowser, Texteditor, Packer (WinZip, WinRAR, unzip o.ä.) und ggf. Linux-Shell etc. muss vorrausgesetzt werden! Wenn Sie sich also nur wenig mit Computern auskennen, sollten Sie sich auf jeden Fall gut damit vertraut machen bevor Sie den Robot Arm in Betrieb nehmen! Eine Einführung in die Bedienung von Computern ist nicht Ziel dieser Anleitung und würde den Rahmen bei weitem sprengen! Hier geht es nur um den Robot Arm, dessen Programmierung und die speziell dafür benötigte Software.

Die Robot Arm CD-ROM
Sie haben vermutlich die Robot Arm CD-ROM im Laufwerk Ihres Computers – falls doch nicht, legen Sie diese nun bitte ein! Es sollte unter Windows kurz darauf per Autostart das CD Menü erscheinen. Andernfalls können Sie über einen Dateimanager die Datei “start.htm” im Hauptverzeichnis der CD mit einem Webbrowser wie z.B. Firefox öffnen. Die Installationsdateien für Firefox finden Sie übrigens auf der CD im Ordner

<CD-ROM-Laufwerk>:\Software\Firefox

sofern Sie noch keinen aktuellen Webbrowser installiert haben sollten. (Es sollte mindestens Firefox 1.x oder der Internet Explorer 6 sein...)

Nach Auswahl der Sprache finden Sie im CD Menü neben dieser Anleitung (die es auch zum Download auf unserer Homepage gibt), vielen Informationen, Datenblättern und Fotos auch den Menüpunkt “Software”. Hier sind alle Software Tools, der USB Treiber und die Beispielprogramme mit Quellcode für den Robot Arm zu finden.

Abhängig von den Sicherheitseinstellungen Ihres Webbrowsers können Sie die Installations-programme direkt von der CD starten!

Wenn Ihr Browser dies aufgrund der Sicherheitseinstellungen nicht erlaubt, müssen Sie die Dateien zunächst in ein Verzeichnis auf Ihrer Festplatte speichern und dann von dort starten. Genaueres dazu steht auf der Software Seite des CD Menüs. Alternativ können Sie natürlich auch direkt in einem Dateimanager auf das CD-Laufwerk wechseln und die Software von dort installieren. Die Verzeichnisnamen sind so gewählt, dass sie eindeutig den entsprechenden Softwarepaketen und Betriebssystemen zugeordnet werden können.

WinAVR - für Windows

Als erstes werden wir WinAVR installieren. WinAVR ist aber - wie der Name schon andeutet- nur für Windows verfügbar !

Linux Anwender müssen beim nächsten Abschnitt weiterlesen.
WinAVR (das wird wie das englische Wort "whenever" ausgesprochen) ist eine Sammlung von vielen nützlichen und notwendigen Programmen für die Software Entwicklung für AVR Mikrocontroller in der Sprache C. WinAVR enthält neben dem GCC für AVR (das nennt sich dann insgesamt "AVR-GCC", mehr Infos dazu folgen später) auch den komfortablen Quelltexteditor "Programmers Notepad 2", den wir auch für die Programmentwicklung für den Robot Arm ein setzen werden!

WinAVR ist ein privat organisiertes Projekt, hinter dem keine Firma o.ä. steht - es ist kostenlos im Internet verfügbar. Neuere Versionen und weitere Informationen finden Sie hier:

http://winavr.sourceforge.net/

Inzwischen wird das Projekt aber auch offiziell von ATMEL unterstützt, und der AVRGCC lässt sich in AVRStudio, die Entwicklungsumgebung für AVRs von ATMEL, einbinden. Das werden wir in diesem Handbuch aber nicht beschreiben, für unsere Zwecke ist Programmers Notepad besser geeignet.
Die WinAVR Installationsdatei finden Sie auf der CD im Ordner:

<CD-ROM-Laufwerk>:\Software\AVR-GCC\Windows\WinAVR\

Die Installation von WinAVR ist sehr einfach und selbsterklärend - normalerweise brauchen keinerlei Einstellungen geändert werden – also einfach immer auf "Weiter" klicken!

Wenn Sie Windows Vista oder Windows 7 benutzen, müssen Sie auf jeden Fall die neueste Version von WinAVR verwenden! Auch mit Windows 2k und XP sollte es problemlos klappen. Falls nicht, können Sie eine der beiden älteren Versionen ausprobieren, die ebenfalls auf der CD zu finden sind (vor Neuinstallation immer bereits installierte WinAVR Versionen wieder deinstallieren!). Offiziell wird Win x64 noch nicht unterstützt, aber auf der CD findet sich ein Patch für Win x64 Systeme falls es Probleme geben sollte. Mehr Infos dazu finden Sie auf der Software Seite des CD Menüs!

AVR-GCC, avr-libc und avr-binutils - für Linux

(Windows Anwender können diesen Abschnitt überspringen!)

Unter Linux kann es schon ein wenig aufwändiger werden. Bei einigen Distributionen sind die benötigten Pakete zwar schon vorhanden, aber meist nur veraltete Versionen. Deshalb müssen Sie neuere Versionen kompilieren und einrichten. Wir können hier nicht im Detail auf jede der zahlreichen Linux Distributionen wie SuSE, Ubuntu, RedHat/Fedora, Debian, Gentoo, Slackware, Mandriva etc. pp. in zig verschiedenen Versionen mit ihren jeweiligen Eigenheiten eingehen und beschreiben das daher nur allgemein.

Das gilt auch für alle anderen Linux Abschnitte in diesem Kapitel!

Das hier beschriebene Vorgehen muss also bei Ihnen nicht unbedingt zum Erfolg führen. Oft kann es hilfreich sein, wenn Sie im Internet z.B. nach "<LinuxDistribution> avr gcc" o.ä. suchen (Verschiedene Schreibweisen ausprobieren). Auch das gilt für alle anderen Linux Abschnitte - natürlich mit angepassten Suchbegriffen! Falls Sie Probleme bei der Installation des AVR-GCC haben, können Sie auch mal in unserem oder im Roboternetz Forum nachschauen bzw. in einem der zahlreichen Linux Foren. Zunächst müssen Sie evtl. schon installierte Versionen des avr-gcc, der avr-binutils und der avr-libc deinstallieren – wie schon gesagt sind diese meist veraltet. Das können Sie mit dem jeweiligen Paketmanager ihrer Distribution tun indem Sie nach „avr" Inbetriebnahme suchen und die drei oben genannten Pakete deinstallieren – sofern diese überhaupt vorhanden sind. Ob der avr-gcc schon installiert ist oder nicht und wenn ja wo, können Sie über eine Konsole z.B. leicht mit

```
> which avr-gcc
```

herausfinden. Sollte hier ein Pfad angezeigt werden, ist schon eine Version installiert. Geben Sie in diesem Fall einfach mal:

```
> avr-gcc --version
```

ein und schauen Sie sich die Ausgabe an. Sollte eine Versionsnummer kleiner als 3.4.6 angezeigt werden, müssen Sie diese alte Version auf jeden Fall deinstallieren.

Wenn die Versionsnummer zwischen 3.4.6 und 4.1.0 liegt, können Sie erstmal versuchen ob Sie Programme kompilieren können (s. nachfolgende Kapitel) und erst wenn das fehlschlägt,die neuen Tools installieren. Wir installieren im Folgenden die derzeit aktuelle Version 4.1.1 (Stand von März 2007) mit einigen wichtigen Patches.

Werden die oben genannten Pakete nicht im Paketmanager angezeigt, obwohl definitiv schon ein avr-gcc vorhanden ist, müssen Sie die entsprechenden Binärdateien manuell löschen – also die /bin, /usr/bin usw. Verzeichnisse nach allen Dateien, die mit „avr-" anfangen absuchen, und diese dann löschen (natürlich NUR diese Dateien und sonst nichts anderes!). Eventuell vorhandene Verzeichnisse wie /usr/avr oder /usr/local/ avr müssen ebenfalls gelöscht werden.

Achtung: Sie müssen unbedingt sicherstellen, dass die normalen Linux Entwicklungstools wie GCC, make, binutils, libc, etc. installiert sind, bevor Sie mit dem Übersetzen und der Installation beginnen können! Das tun Sie am besten über den Paketmanager Ihrer Distribution. Jede Linux Distribution sollte die benötigten Pakete schon auf der Installations CD mitliefern bzw. aktuelle Pakete über das Internet bereitstellen.

Stellen Sie sicher, dass das Programm „texinfo" installiert ist. Installieren Sie bitte ggf. das entsprechende Paket, bevor Sie weitermachen – sonst klappt es nicht!

Ist das erledigt, kann mit der eigentlichen Installation begonnen werden.

Es gibt nun zwei Möglichkeiten, entweder man macht alles von Hand, oder man nutzt ein sehr einfach anzuwendendes Installationsskript.

Wir empfehlen es zunächst mit dem Skript zu versuchen. Wenn das nicht klappt, kann man immer noch den Compiler von Hand einrichten!

Achtung:
Sie sollten für die Installation noch genug freien Speicherplatz auf der Festplatte zur Verfügung haben! Temporär werden mehr als 400MB benötigt. Über 300MB davon können nach der Installation wieder gelöscht werden, aber während der Übersetzung braucht man den Platz.

Viele der nachfolgenden Installationsschritte erfordern **ROOT RECHTE**, also loggen Sie sich ggf. mit „su" als root ein oder führen Sie die kritischen Befehle mit „sudo" o.ä. aus, wie man es z.B. bei Ubuntu machen muss (das Installationsskript, mkdir in /usr/local Verzeichnissen und make install brauchen root Rechte).

Achten Sie im Folgenden bitte auf EXAKTE Schreibweise aller Befehle!
Jedes Zeichen ist wichtig und auch wenn einige Befehle evtl. etwas seltsam aussehen – das ist alles richtig so und kein Tippfehler!
(<CD-ROM-Laufwerk> muss man natürlich trotzdem mit dem Pfad des CD-ROM-Laufwerks ersetzen!)

Alle für uns relevanten Installationsdateien für den avr-gcc, avr-libc und binutils finden Sie auf der CD im Ordner:

<CD-ROM-Laufwerk>:\Software\avr-gcc\Linux

Zunächst müssen Sie alle Installationsdateien in ein Verzeichnis auf Ihrer Festplatte kopieren – **das gilt für beide Installationsvarianten!** Hier nutzen wir das Home Verzeichnis (übliche Abkürzung für das aktuelle Home Verzeichnis ist die Tilde: „~"):

```
> mkdir ~/Robot Arm
> cd <CD-ROM-Laufwerk>/Software/avr-gcc/Linux
> cp * ~/Robot Arm
```

Die Dateien können Sie nach der erfolgreichen Installation natürlich wieder löschen um Platz zu sparen!

Automatisches Installationsskript

Wenn man das Skript mit chmod ausführbar gemacht hat, kann es sofort losgehen:

```
> cd ~/Robot Arm
> chmod -x avrgcc_build_and_install.sh
> ./avrgcc_build_and_install.sh
```

Die Nachfrage, ob man mit dieser Konfiguration installieren möchte oder nicht, können Sie mit „y“ beantworten.

ACHTUNG: Das Übersetzen und Installieren wird dann je nach Rechenleistung Ihres Systems einige Zeit in Anspruch nehmen. (z.B. etwa 15 min auf einem 2GHz CoreDuo Notebook – bei langsameren Systemen evtl. entsprechend länger)

Das Skript spielt auch einige Patches ein – das sind diese ganzen .diff Dateien, die in dem Verzeichnis liegen.
Wenn alles klappt, sollte ganz zum Schluss folgendes erscheinen:

```
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) installation of avr GNU tools complete
(./avrgcc_build_and_install.sh) add /usr/local/avr/bin to your path to use the avr GNU tools
(./avrgcc_build_and_install.sh) you might want to run the following to save disk space:
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) rm -rf /usr/local/avr/source /usr/local/avr/build
```

Dann können Sie wie es dort vorgeschlagen wird
rm -rf /usr/local/avr/source /usr/local/avr/build
ausführen! Das löscht alle temporären Dateien, die Sie normalerweise nicht mehr benötigen.

Jetzt können Sie den nächsten Abschnitt überspringen und noch den Pfad auf die avr tools setzen.

Sollte die Ausführung des Skriptes fehlschlagen, müssen Sie sich genau die Fehlermeldungen ansehen (auch mal in der Konsole etwas hochscrollen) – meist fehlen dann irgendwelche Programme, die man vorher noch installieren muss (wie z.B. das oben erwähnte texinfo). Bevor Sie nach einem Fehler weitermachen, sollten Sie die bereits erzeugten Dateien im Standardinstallationsverzeichnis „/usr/local/avr“ vorsichtshalber löschen – am besten das ganze Verzeichnis.

Wenn Sie nicht wissen, was da genau falsch gelaufen ist, bitte alle Kommando-zeilenausgaben in einer Datei speichern und damit an den Support wenden. Bitte immer so viele Informationen wie möglich mitsenden! Dann wird es einfacher, Ihnen zu helfen.

GCC für den AVR

Der GCC wird ähnlich wie die Binutils gepatcht, übersetzt und installiert:

```
> cd ~/Robot Arm> bunzip2 -c gcc-4.1.1.tar.bz2 | tar xf -
> cd gcc-4.1.1
> patch -p0 < ../gcc-patch-0b-constants.diff
> patch -p0 < ../gcc-patch-attribute_alias.diff
> patch -p0 < ../gcc-patch-bug25672.diff
> patch -p0 < ../gcc-patch-dwarf.diff
> patch -p0 < ../gcc-patch-libiberty-Makefile.in.diff
> patch -p0 < ../gcc-patch-newdevices.diff
> patch -p0 < ../gcc-patch-zz-atmega256x.diff
> mkdir obj-avr
> cd obj-avr
> ../configure --prefix=$PREFIX --target=avr --enable-languages=c,c++ \
--disable-nls --disable-libssp –with-dwarf2
> make
> make install
```

Nach dem \ kann man einfach Enter drücken und weiterschreiben – so kann der Befehl auf mehrere Zeilen aufgeteilt werden. Kann man aber auch ganz weglassen.

AVR Libc

Und schließlich noch die AVR libc:

```
> cd ~/Robot Arm
> bunzip2 -c avr-libc-1.4.5.tar.bz2 | tar xf -
> cd avr-libc-1.4.5
> ./configure --prefix=$PREFIX --build=`./config.guess` --host=avr
> make
> make install
```

Achtung: bei –build=`./config.guess` darauf achten auch den „Accent grave“ (à <-- den Strich da auf dem a! Neben der Backspace Taste – rechts oben auf der Tastatur, einmal mit Shift diese Taste drücken und danach die Leertaste) und kein normales Hochkomma oder Anführungszeichen zu benutzen, sonst klappt es nicht.

Pfad setzen

Sie müssen jetzt dafür sorgen, dass das Verzeichnis /usr/local/avr/bin auch in der Pfad Variablen eingetragen ist – sonst kann man den avr-gcc nicht aus der Konsole bzw. aus Makefiles heraus aufrufen. Dazu müssen Sie den Pfad in die Datei /etc/profile bzw. /etc/environment o.ä. (variiert von Distribution zu Distribution) eintragen – mit einem Doppelpunkt „:“ getrennt von den anderen schon vorhandenen Einträgen. In der Datei könnte das dann in etwa so aussehen:

PATH=”/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/local/avr/bin“

Jetzt in einer beliebigen Konsole „avr-gcc —version“ eingeben wie weiter oben beschrieben – wenn das funktioniert, ist die Installation gelungen!

Manuelle Installation

Wenn Sie den Compiler lieber von Hand einrichten wollen oder die Installation mit dem Skript nicht klappt, können Sie nach den Anweisungen im folgenden Abschnitt vorgehen.

Die Beschreibung hier orientiert sich an diesem Artikel:

http://www.nongnu.org/avr-libc/user-manual/install_tools.html

der auch in der AVR Libc Dokumentation im PDF Format auf der CD zu finden ist:

<CD-ROM-Laufwerk>:\Software\Documentation\avr-libc-user-manual-1.4.5.pdf

Wir fassen uns hier zwar sehr viel kürzer, spielen aber gleich noch ein paar wichtige Patches ein – ohne diese funktionieren einige Dinge nicht richtig.

Zunächst müssen wir uns ein Verzeichnis erstellen, in das wir alle Tools installieren werden. Das sollte /usr/local/avr sein.
Also in einer Konsole ALS ROOT folgendes eingeben:

```
> mkdir /usr/local/avr
> mkdir /usr/local/avr/bin
```

Es muss nicht unbedingt dieses Verzeichnis sein. Wir legen dafür einfach die Variable $PREFIX für dieses Verzeichnis an:

```
> PREFIX=/usr/local/avr
> export PREFIX
```

Das muss nun noch unbedingt der PATH Variable hinzugefügt werden:

```
> PATH=$PATH:$PREFIX/bin
> export PATH
```

Binutils für AVR

Nun müssen Sie den Quellcode der Binutils entpacken und ein paar Patches einspielen.
Wir nehmen hier an, dass Sie alles ins Home Verzeichnis ~/Robot Arm kopiert haben:

```
> cd ~/Robot Arm
> bunzip2 -c binutils-2.17.tar.bz2 | tar xf -
> cd binutils-2.17
> patch -p0 < ../binutils-patch-aa.diff
> patch -p0 < ../binutils-patch-atmega256x.diff
> patch -p0 < ../binutils-patch-coff-avr.diff
> patch -p0 < ../binutils-patch-newdevices.diff
> patch -p0 < ../binutils-patch-avr-size.diff
> mkdir obj-avr
> cd obj-avr
```

Nun wird das configure Skript ausgeführt:

```
> ../configure --prefix=$PREFIX --target=avr --disable-nls
```

Dieses Skript ermittelt, was auf Ihrem System verfügbar ist und erzeugt dementsprechend passende Makefiles. Jetzt können die Binutils übersetzt und installiert werden:

```
> make
> make install
```

Das kann je nach Rechenleistung Ihres Systems schon ein paar Minuten dauern – das gilt auch für die beiden nächsten Abschnitte – vor allem für den GCC!

Java 6

Der RobotLoader (Infos dazu s.u.) wurde für die Java Plattform entwickelt und ist unter Windows und Linux verwendbar (theoretisch auch andere Betriebsysteme wie OS X, aber hier kann AREXX Engineering leider noch keinen offiziellen Support leisten). Damit das funktioniert, ist es notwendig, ein aktuelles Java Runtime Environment (JRE) zu installieren. Oft haben Sie dies bereits auf dem Rechner, allerdings muss es mindestens Version 1.6 (= Java 6) sein! Falls Sie also noch kein JRE oder JDK installiert haben, müssen Sie zunächst das auf der CD mitgelieferte JRE 1.6 der Firma SUN Microsystems installieren, oder alternativ eine neuere Version von http://www.java.com oder http://java.sun.com downloaden.

Windows

Das JRE 1.6 befindet sich für Windows in folgendem Ordner:

<CD-ROM-Laufwerk>:\Software\Java\JRE6\Windows\

Unter Windows ist die Installation von Java sehr einfach - Sie müssen nur den Setup starten und den Anweisungen auf dem Bildschirm folgen - fertig. Den nächsten Abschnitt können Sie überspringen.

Linux

Unter Linux ist die Installation meistens auch relativ problemlos möglich, bei einigen Distributionen kann es aber ein wenig Handarbeit erfordern.

In diesem Ordner:

<CD-ROM-Laufwerk>:\Software\Java\JRE6\

finden Sie das JRE1.6 als RPM (SuSE, RedHat etc.) und als selbstextrahierendes Archiv „.bin". Unter Linux ist es besser wenn Sie zunächst im Paketmanager Ihrer jeweiligen distribution nach Java Paketen suchen (Suchbegriffe z.B. „java", „sun", „jre", „java6" ...) und dann diese distributionseigenen Pakete verwenden und nicht die auf dieser CD-ROM! Achten Sie aber unbedingt darauf Java 6 (= 1.6) oder ggf. eine neuere Version zu installieren und keine ältere Version!

Unter Ubuntu oder Debian funktioniert das RPM Archiv nicht direkt – hier müssen Sie die Paketmanager Ihrer jeweiligen Distribution bemühen, um an ein passendes Installationspaket zu kommen. Das RPM sollte bei vielen anderen Distributionen wie RedHat/Fedora und SuSE aber problemlos funktionieren. Falls nicht, bleibt noch der Weg das JRE aus dem selbstextrahierenden Archiv (.bin) zu entpacken (z.B.nach /usr/lib/Java6) und dann manuell die Pfade zum JRE zu setzen (PATH undJAVA_HOME etc.).

Bitte beachten Sie hier auch die Installationsanweisungen von Sun – die ebenfalls im oben genannten Verzeichnis und auf der Java Website (s.o.) zu finden sind!

Ob Java korrekt installiert wurde, können Sie in einer Konsole überprüfen, indem Sie den Befehl „java -version“ ausführen. Es sollte in etwa folgende Ausgabe erscheinen:

```
java version “1.6.0”
Java(TM) SE Runtime Environment (build 1.6.0-b105)
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

Steht dort etwas ganz anderes, haben Sie entweder die falsche Version installiert, oder auf Ihrem System ist noch eine andere Java VM installiert.

Robot Loader

Der Robot Loader wurde entwickelt, um komfortabel neue Programme in den Robot Arm und alle Erweiterungsmodule laden zu können (sofern diese über einen Mikrocontroller mit kompatiblem Bootloader verfügen). Weiterhin sind ein paar nützliche Zusatzfunktionen integriert, wie z.B. ein einfaches Terminal-programm.

Den Robot Loader selbst braucht man nicht zu installieren – das Programm kann einfach irgendwo in einen neuen Ordner auf die Festplatte kopiert werden. Der Robot Loader befindet sich in einem Zip-Archiv auf der CD-ROM:

<CD-ROM-Laufwerk>:\Software\RobotLoader\RobotLoader.zip

Dieses müssen Sie nur irgendwo auf die Festplatte entpacken – z.B. in einen neuen Ordner C:\Programme\RobotLoader (o.ä.). In diesem Ordner finden Sie dann die Datei RobotLoader.exe und können sie mit einem Doppelklick starten.

Das eigentliche Robot Loader Programm liegt im Java Archive (JAR) RobotLoader_lib.jar. Dieses können Sie alternativ auch von der Kommandozeile aus starten.

Unter Windows:

```
java -Djava.library.path=”.\lib” -jar RobotLoader_lib.jar
```

Linux:

```
java -Djava.library.path=”./lib” -jar RobotLoader_lib.jar
```

Diese lange -D Option ist notwendig, damit die JVM auch alle verwendeten Bibliotheken finden kann. Unter Windows braucht man das aber nicht und kann einfach die .exe Datei zum Starten verwenden und für Linux gibt es ein Shell Skript „RobotLoader. sh“. Das Skript muss evtl. zunächst noch ausführbar gemacht werden (chmod -x ./RobotLoader.sh). Danach kann man es in einer Konsole mit „./RobotLoader.sh“ starten.

Es empfiehlt sich, eine Verknüpfung auf dem Desktop oder im Startmenü anzulegen, um den Robot Loader bequem starten zu können. Unter Windows geht das z.B. einfach indem man rechts auf die Datei RobotLoader.exe klickt und dann im Menü „Senden an“ auf „Desktop (Verknüpfung erstellen)“ klickt.

Robot Arm Library, Robot Arm CONTROL Library und Beispielprogramme

Die Robot Arm Library und die zugehörigen Beispielprogramme befinden sich in einem Zip-Archiv auf der CD:

<CD-ROM-Laufwerk>:\Software\Robot Arm Examples\Robot ArmExamples **[PRO]**.zip

Sie können diese einfach direkt in ein Verzeichnis Ihrer Wahl auf die Festplatte entpacken. Am besten entpacken Sie die Beispielprogramme in einen Ordner auf einer Daten Partition. Oder in den „Eigene Dateien“ Ordner in einem Unterordner „Robot Arm\Examples\“ bzw. unter Linux ins Home Verzeichnis. Das steht Ihnen aber völlig frei.

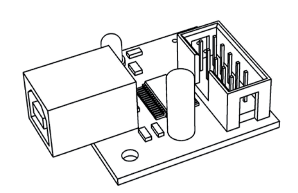
Die einzelnen Beispielprogramme werden noch später im Softwarekapitel besprochen!

7. Programmer und Loader

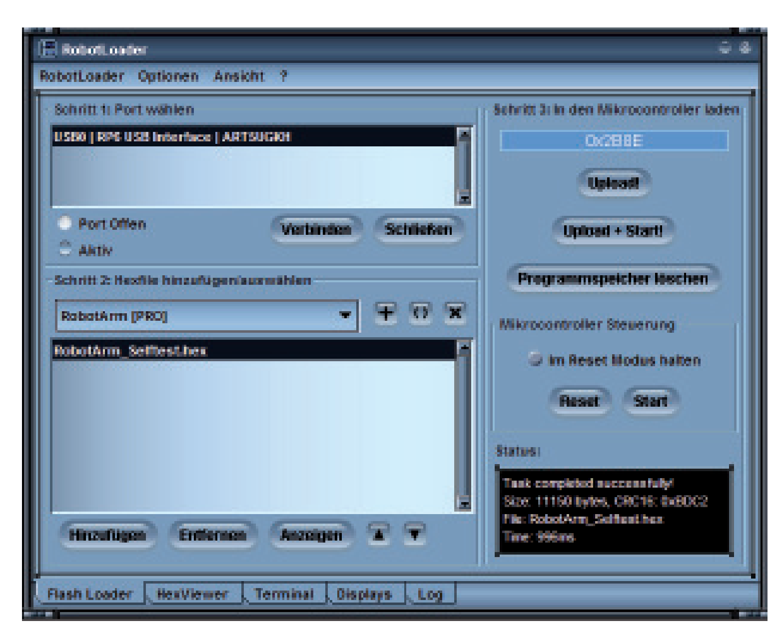
Zum Laden eines beliebigen HEX Roboter Arm-Programms vom PC in den Roboter Arm verwenden wir den USB Programmier adapter und unsere RobotLoader Software.

Der im Lieferumfang enthaltene lose USB-Port-Adapter Sender/Empfänger (Transceiver) wird einerseits an einem USB-Port des Computers angeschlossen und andererseits an den Prog/UART- Port der Roboterarm-Platine.

Der Ladevorgang eines Programms in den Roboter Arm überschreibt automatisch das bereits zuvor vorhandene Programm.



USB Programmier adapter



RobotLoader software

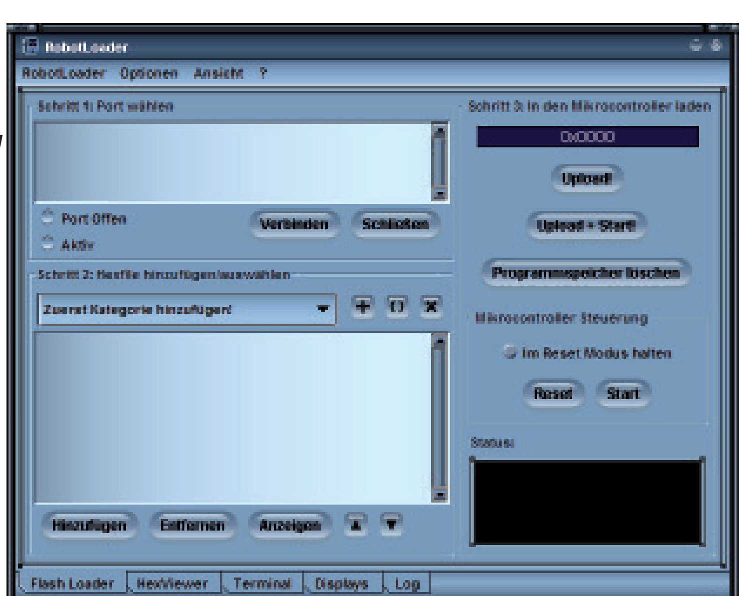
7.1. Robot Loader

Wie bereits gesagt wurde der RobotLoader entwickelt um komfortabel neue Programme in den Roboter Arm und alle unsere Roboter übertragen (sofern diese über einen Mikrocontroller mit kompatiblem Bootloader verfügen).

RobotLoader

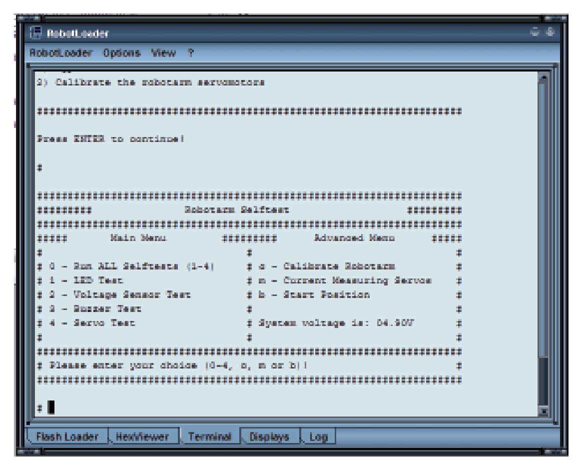
Wann die Spannung < 4,4 Volt, erscheint eine Warnung.

Die maximale Spannung welche der RobotLoader messen kan ist 5,1 Volt!



Weiterhin sind ein paar nützliche Zusatzfunktionen integriert, wie z.B. ein einfaches Terminalprogramm.

Terminal Fenster



Den RobotLoader selbst braucht man nicht installieren – man kann das Programm einfach irgendwo in einen neuen Ordner auf die Festplatte kopieren..

7.2. Anschluss des USB Interfaces – Windows

Linux Anwender können beim nächsten Abschnitt weiterlesen!

Zur Installation des USB Interfaces gibt es mehrere Möglichkeiten. Die einfachste Möglichkeit ist es, d**en Treiber VOR dem ersten Anschließen des Geräts zu installieren.**

Auf der CD befindet sich ein Installationsprogramm für den Treiber.

Für **32 und 64 Bit Windows 7, XP, Vista, Server 2003 und 2000** Systeme:

<CD-ROM-Laufwerk>:\Software\USB_DRIVER\Win2k_XP\CDM_Setup.exe

Für alte **Win98SE/Me** Systeme gibt es so ein komfortables Programm leider nicht. Hier muss ein älterer Treiber von Hand installiert werden nachdem man das Gerät angeschlossen hat (s.u.).

Das CDM Installationsprogramm müssen Sie einfach nur ausführen – es gibt nur eine kurze Rückmeldung, sobald der Treiber installiert wurde, sonst passiert nichts weiter.

Dann können Sie das USB Interface an den PC anschließen. **BITTE NOCH NICHT MIT DEM ROBOTER VERBINDEN!** Einfach nur über das USB Kabel mit dem PC verbinden! Dabei sollten Sie darauf achten, die Platine des USB Interfaces nur am Rand oder am USB Stecker bzw. an der Kunststoffwanne des Programmiersteckers anzufassen (s. Sicherheitshinweise zu statischen Entladungen)! Sie sollten besser keine der Bauteile auf der Platine, Lötstellen oder die Kontakte des Wannensteckers berühren, wenn nicht unbedingt nötig, um statische Entladungen zu vermeiden!

Der zuvor installierte Treiber wird nun automatisch für das Gerät verwendet, ohne dass Sie noch etwas zu tun brauchen. Es erscheinen bei Windows XP/2k kleine Sprechblasen unten über der Taskleiste – die letzte Meldung sollte in etwa „Die Hardware wurde installiert und kann nun verwendet werden!“ lauten!

Wenn Sie das USB Interface doch schon vor der Installation angeschlossen haben (oder Win98/Me benutzen) – auch nicht schlimm. Dann werden Sie von Windows nach einem Treiber gefragt. Auch diese Installationsvariante ist möglich, der Treiber befindet sich auch in entpackter Form auf der CD!

Wenn dies bei Ihnen der Fall ist, erscheint (unter Windows) für gewöhnlich ein Dialog zum Installieren eines neuen Gerätetreibers. Sie müssen dem System dann den Pfad angeben, unter dem es den Treiber finden kann. Bei Windows 2k/XP muss man erst auswählen, den Treiber manuell zu installieren und natürlich keinen Webdienst o.ä. zu suchen. Der Treiber befindet sich in unserem Fall auf der CD in den oben genannten Verzeichnissen.

Also einfach das jeweilige Verzeichnis für Ihre Windows Version angeben und evtl. noch ein paar Dateien, die das System nicht selbstständig findet (sind alle in den weiter unten genannten Verzeichnissen!)...

Bei Windows XP oder späteren Versionen folgt oft (hier normalerweise nicht, da die FTDI Treiber signiert sind) noch ein Hinweis das der Treiber nicht von Microsoft signiert/verifiziert worden ist - das ist irrelevant und kann bedenkenlos bestätigt werden.

Inbetriebnahme

Für **32 und 64 Bit Windows 7, XP, Vista, Server 2003 und 2000 Systeme:**

<CD-ROM-Laufwerk>:\Software\USB_DRIVER\Win2k_XP\FTDI_CDM2\

Für ältere **Windows 98SE/Me Systeme**:

<CD-ROM-Laufwerk>:\Software\USB_DRIVER\Win98SE_ME\FTDI_D2XX\

Bei einigen älteren Windows Versionen wie Win98SE ist evtl. nach Installation des Treibers ein Neustart erforderlich! ACHTUNG: Unter Win98/Me funktioniert nur einer von beiden Treibern: Virtual Comport oder der D2XX Treiber von FTDI! Hier gibt es leider keinen Treiber, der beide Funktionen integriert und es steht normalerweise kein virtueller Comport zur Verfügung, da der RP6Loader unter Windows standardmäßig die D2XX Treiber verwendet (das kann man auch ändern - kontaktieren Sie hier ggf.unser Support Team!).

Überprüfen, ob das Gerät richtig angeschlossen ist

Um zu überprüfen ob das Gerät korrekt installiert worden ist, kann man unter Windows XP, 2003 und 2000 neben dem Robot Loader auch den Gerätemanager verwenden: Rechtsklick auf den Arbeitsplatz --> Eigenschaften --> Hardware --> Gerätemanager

ODER alternativ: Start --> Einstellungen --> Systemsteuerung --> Leistung und Wartung --> System --> Hardware --> Gerätemanager und dort in der Baumansicht unter “Anschlüsse (COM und LPT)” nachsehen ob ein “USB-Serial Port (COMX)” zu sehen ist - wobei das X für die Portnummer steht oder unter „USB-Controller“ nach einem „USB Serial Converter“ suchen!

Treiber später wieder Deinstallieren

Sollten Sie den Treiber jemals wieder deinstallieren wollen (Nein, das tun Sie jetzt bitte nicht - ist nur ein Hinweis falls Sie das jemals brauchen sollten): Wenn Sie das CDM Installationsprogramm verwendet haben, können Sie das direkt über Start-->Einstellungen--> Systemsteuerung-->Software tun. In der dortigen Liste finden Sie einen Eintrag des „FTDI USB Serial Converter Drivers“ – diesen auswählen und dort dann auf deinstallieren klicken!

Wenn Sie den Treiber von Hand installiert haben, können Sie das Programm “FTUNIN.exe” im Verzeichnis des jeweiligen USB Treibers für Ihr System ausführen! Achtung: USB-->RS232 Adapter mit FTDI Chipsatz verwenden meist ebenfalls diesen Treiber!

7.3. Anschluss des USB Interfaces – Linux

Windows Anwender können diesen Abschnitt überspringen!

Bei Linux mit Kernel 2.4.20 oder höher ist der benötigte Treiber schon vorhanden (zumindest für das kompatible Vorgängermodell FT232BM des Chips auf unserem USB Interface, dem FT232R), das Gerät wird automatisch erkannt und Sie brauchen nichts weiter zu tun. Falls es doch mal Probleme gibt, erhalten Sie Linux Treiber (und Support und auch evtl. neuere Treiber) direkt von FTDI:

http://www.ftdichip.com/

Unter Linux kann man, nachdem man das Gerät angeschlossen hat, mit:
cat /proc/tty/driver/usbserial

anzeigen lassen, ob der USB-Serial Port korrekt installiert worden ist. Mehr braucht man hier normalerweise nicht zu tun.

Allerdings sei noch darauf hingewiesen, dass der Robot Loader unter Windows die D2XX Treiber verwendet und dort die vollständigen USB Bezeichnungen in der Portliste auftauchen (z.B. „USB0 | Robot USB Interface | serialNumber“). Unter Linux sind es stattdessen die virtuellen Comport Bezeichnungen /dev/ttyUSB0, /dev/ttyUSB1 etc.. Es werden auch die normalen Comports angezeigt, als „dev/ttyS0“ usw.. Hier müssen Sie ausprobieren welcher Port der richtige ist!

Für Linux ist leider kein so einfach zu installierender Treiber verfügbar der beides bereitstellt. Von daher war es hier sinnvoller die Virtual Comport Treiber, die ohnehin schon im Kernel vorhanden sind, zu verwenden. Die D2XX Treiber würden bei der Installation auch noch einiges an Handarbeit erfordern...

Software Installation abschließen

Das war es auch schon mit der Installation der Software und des USB Interfaces! Jetzt könnten Sie noch die wichtigsten Dateien von der CD auf die Festplatte kopieren (vor allem den kompletten Ordner „Documentation“ und, falls nicht schon geschehen, die Beispielprogramme). Dann müssen Sie nicht ständig die CD suchen, wenn Sie diese Dateien benötigen! Die Ordner auf der CD sind alle so benannt, dass sie eindeutig den jeweiligen Softwarepaketen bzw. der jew. Dokumentation zugeordnet werden können!

Sollten Sie die CD einmal “verlegen”, können Sie die wichtigsten Dateien wie dieses Handbuch, den Robot Loader und die Beispielprogramme auch von der AREXX Homepage downloaden. Dort finden Sie auch Links zu den anderen Softwarepaketen, die Sie benötigen.

7.4. USB Interface Testen und RobotLoader starten

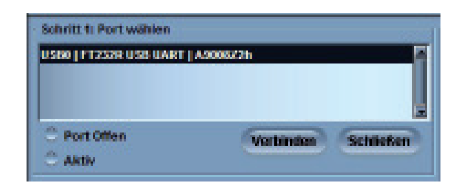
Als nächstes testen wir den Programmupload über das USB Interface. Verbinden Sie bitte das USB Interface mit dem PC (Immer zuerst mit dem PC verbinden!) und danach über das 10-pol. Flachbandkabel mit dem “PROG/UART” Anschluss des Robot Arm! (Robot Arm MUSS ABGESCHALTET SEIN!) Das 10-pol. Flachbandkabel ist mechanisch gegen Verpolung geschützt, sofern man es also nicht mit Gewalt behandelt, kann man es gar nicht verkehrt herum anschließen.



Starten Sie danach den RobotLoader. Je nachdem welche Sprache Sie gewählt haben, können die Menüs natürlich etwas anders beschriftet sein. Auf den Screenshots ist die englische Sprachversion dargestellt, über den Menüpunkt „Options->Preferences“ und dann bei „Language /Sprache“ kann man die Sprache anpassen (Englisch oder Deutsch) und danach auf OK klicken.

Nach Änderung der Sprache muss man den RobotLoader aber erst einmal neu starten bevor sich etwas ändert!

Port öffnen - Windows



Jetzt können Sie den USB Port auswählen. Sofern kein anderer USB->Seriell Adapter mit FTDI Controller am PC angeschlossen ist, sehen sie in der Portliste nur einen einzigen Eintrag den Sie dann bitte auswählen.

Falls doch mehrere Ports vorhanden sind, können Sie den Port anhand des Namens „Robot USB Interface“ identifizieren (oder „FT232R USB UART“). Dahinter wird noch die einprogrammierte Seriennummer angezeigt.

Sollten keine Ports angezeigt werden, können sie im Menü über „RobotLoader-->Refresh Portlist“ („RobotLoader-->Portliste aktualisieren“) die Portliste aktualisieren!



ACHTUNG!

Wann die Spannung < 4,4 Volt, erscheint eine Warnung.
Die maximale Spannung welche der RobotLoader messen kan ist 5,1 Volt!

7.5. Port öffnen – Linux

Unter Linux wird der USB-Seriell Adapter wie ein normaler Comport behandelt. Die D2XX Treiberinstallation von FTDI unter Linux wäre nicht ganz so einfach und die normalen Virtual Comport (VCP) Treiber sind in aktuellen Linux Kerneln sowieso schon enthalten. Es funktioniert alles fast genauso wie unter Windows, nur muss man erst noch kurz ausprobieren welchen Namen das Robot Arm USB Interface hat und darauf achten, den USB Port nicht vom PC zu trennen solange die Verbindung noch offen ist (ansonsten muss der RobotLoader eventuell neu gestartet werden damit die Verbindung wieder klappt). Die Virtual Comports heissen unter Linux „/dev/ttyUSBx“, wobei x eine Nummer ist, z.B. „/dev/ttyUSB0“ oder „/dev/ttyUSB1“. Die normalen Comports heissen unter Linux „/dev/ttyS0“, „/dev/tty- S1“ etc.. Diese tauchen ebenfalls in der Portliste auf sofern vorhanden.

Der RobotLoader merkt sich – wenn es denn überhaupt mehrere Ports gibt - welchen Port Sie zuletzt verwendet haben und selektiert diesen bei jedem Start des Programms automatisch (generell bleiben die meisten Einstellungen und Selektionen erhalten).

Nun können Sie auf den Button „Connect“ (=„Verbinden“) klicken! Der RobotLoader öffnet nun den Port und testet, ob die Kommunikation mit dem Bootloader auf dem Roboter funktioniert. Es sollte dann unten im schwarzen Feld „Status“ die Nachricht “Connected to: Robot Arm ...” o.ä. zusammen mit einer Info über die aktuell gemessene Spannung erscheinen. Falls nicht, probieren Sie es nochmal! Wenn es dann immer noch nicht klappt, liegt ein Fehler vor! Schalten Sie in diesem Fall den Roboter am Besten sofort aus und fangen Sie an mit Fehlersuche!



ACHTUNG!

Wann die Spannung < 4,4 Volt, erscheint eine Warnung.
Die maximale Spannung welche der RobotLoader messen kan ist 5,1 Volt!

7.6. SELBSTTEST

Sobald der Roboter Arm eingeschaltet wird leuchtet die Spannung-LED Gelb.

Die Status-LED erlischt beim Upload einer HEX Datei. Sobald wir ein Programm starten leuchtet die Status-LED rot auf und im Roboter-Status „Fertig“ leuchtet diese LED grün.

Wenn das geklappt hat, können Sie ein kleines Selbsttestprogramm ausführen, um die Funktionsfähigkeit aller Systeme des Roboters zu überprüfen. Klicken Sie dazu unten im Robot Loader Fenster auf den Button „Add“ („Hinzufügen“) und wählen Sie die Datei RobotArmExamples [PRO], „Example_11_Selftest\RobotArm_Selftest.hex“ im Beispielverzeichnis aus. In dieser Datei befindet sich das Selbsttestprogramm im hexadezimalen Format – daher werden solche Programmdateien auch „Hexdateien“ genannt. Die eben ausgewählte Datei taucht anschließend in der Liste auf. So können Sie später auch noch andere Hexdateien von Ihren eigenen Programmen und den Beispielprogrammen hinzufügen (s. Screenshot, hier wurden schon ein paar Hexdateien hinzugefügt). Der Robot Loader kann auch verschiedene Kategorien von Hexdateien verwalten.

Damit lassen sich die Dateien übersichtlicher sortieren. Beispielsweise wenn man mehrere programmierbare Erweiterungsmodule auf dem Robot Arm hat, oder verschiedene Varianten von Programmen verwendet. Die Liste wird immer automatisch beim Beenden des Programms gespeichert! Es werden hier natürlich nur die Pfade zu den Hexdateien gespeichert – nicht die Hexdateien selbst. Wenn Sie an einem Programm arbeiten, brauchen Sie die Hexdatei nur einmal hinzuzufügen und auszuwählen, und können danach sofort nach jedem erneuten Übersetzen des Programms, das neue Programm in den Mikrocontroller laden (auch per Tastenkombination [STRG+D] oder [STRG+Y], um das Programm direkt nach dem Übertragen zu starten). Unter verschiedenen Betriebssystemem sind die Pfadnamen natürlich komplett anders. Sie können denRobot Loader trotzdem ohne weitere Änderungen direkt unter Windows und Linux verwenden, denn es gibt für Windows und Linux jeweils eine extra Liste.

Sie können nun entweder mit den anderen Beispielprogrammen (Examples) des Robot Arm weitermachen oder mit Ihrer eigenen Softwareprogrammierung anfangen.

ACHTUNG!

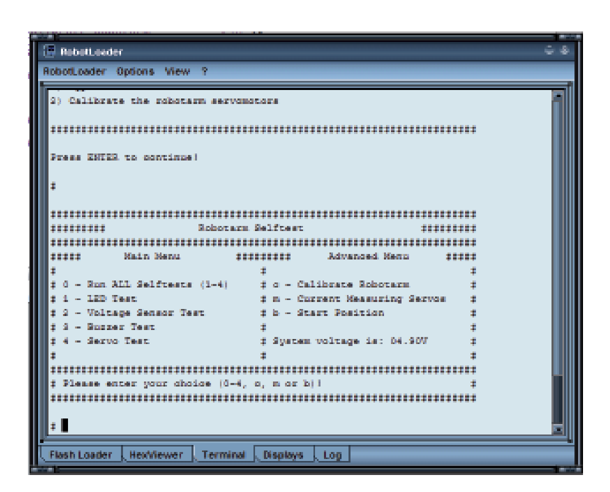
Wann die Spannung < 4,4 Volt, erscheint eine Warnung.

Die maximale Spannung welche der RobotLoader messen kan ist 5,1 Volt!

Wählen Sie dazu die Datei „RobotArm_Selftest.hex“ in der Liste und aktivieren Sie anschließend den Button „Upload!“ rechts oben unterhalb dem Fortschrittsbalken.

Das Programm wird nun in den MEGA64-Prozessor auf dem Roboter Arm übertragen. Das sollte nicht länger als ein paar Sekunden dauern (maximal 5 Sekunden beim Selbsttest Programm).

Wechseln Sie anschließend bitte auf den Karteireiter (=„Tabs“ ganz unten im Programmfenster!) „Terminal“! Alternativ dazu können Sie auch ins Terminal wechseln im Menü „View“ („Ansicht“).



Sie können jetzt den Selbsttest und die Kalibrierung des Roboter Arms durchführen. Starten Sie dazu bitte das Programm mit einem Druck auf den Start/Stop Reset Taster auf dem Roboter ARM Später können Sie das natürlich auch alternativ über das Menü RobotLoader --> Start oder die Tastenkombination [STRG]+[S] tun, so können Sie allerdings direkt ausprobieren ob der Taster korrekt funktioniert!

Falls im Selbsttest ein Fehler auftritt, schalten Sie den Roboter am Besten sofort aus und fangen Sie mit der Fehlersuche an.

AM BESTEN FANGEN SIE MIT DER KALIBRIERUNG DES ROBOTER ARMS AN! SIEHE SEITE 46.

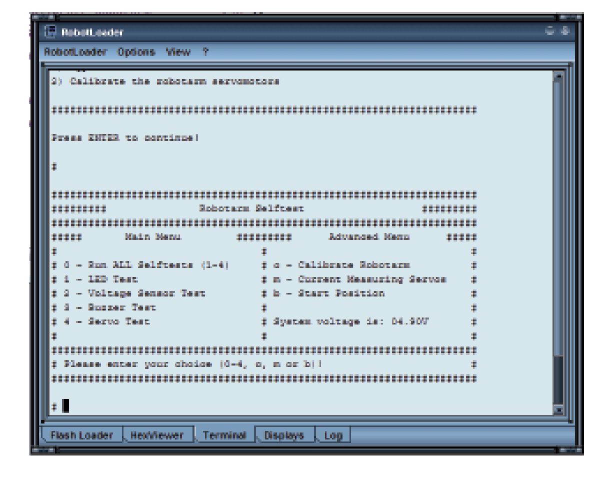
7.7. Kalibrierung

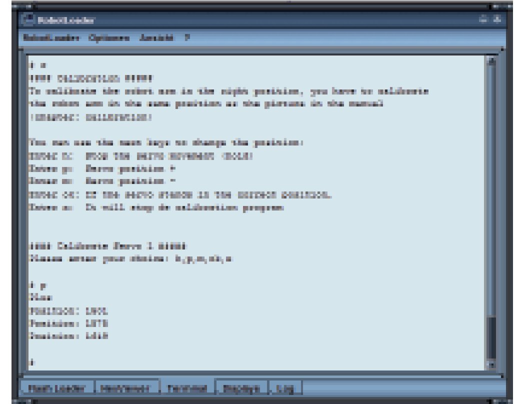
Starten Sie Kalibrierungsprogramm, um die Roboter Arm zu Kalibrieren.

Klicken Sie dazu unten im RobotLoader-Fenster auf den Button „Add“ („Hinzufügen“) und wählen Sie die Datei RobotArmExamples [PRO], „Example_11_Selftest\RobotArm_Selftest.hex“ im Beispielverzeichnis aus.

In dieser Datei befindet sich das Selbsttestprogramm im hexadezimalen Format. Die eben ausgewählte Datei taucht anschließend in der Liste auf. (siehe Screenshot).

Wählen Sie dann C (C - Calibrate) des Kalibrierungsprogramms zum Start der Kalibrierung.

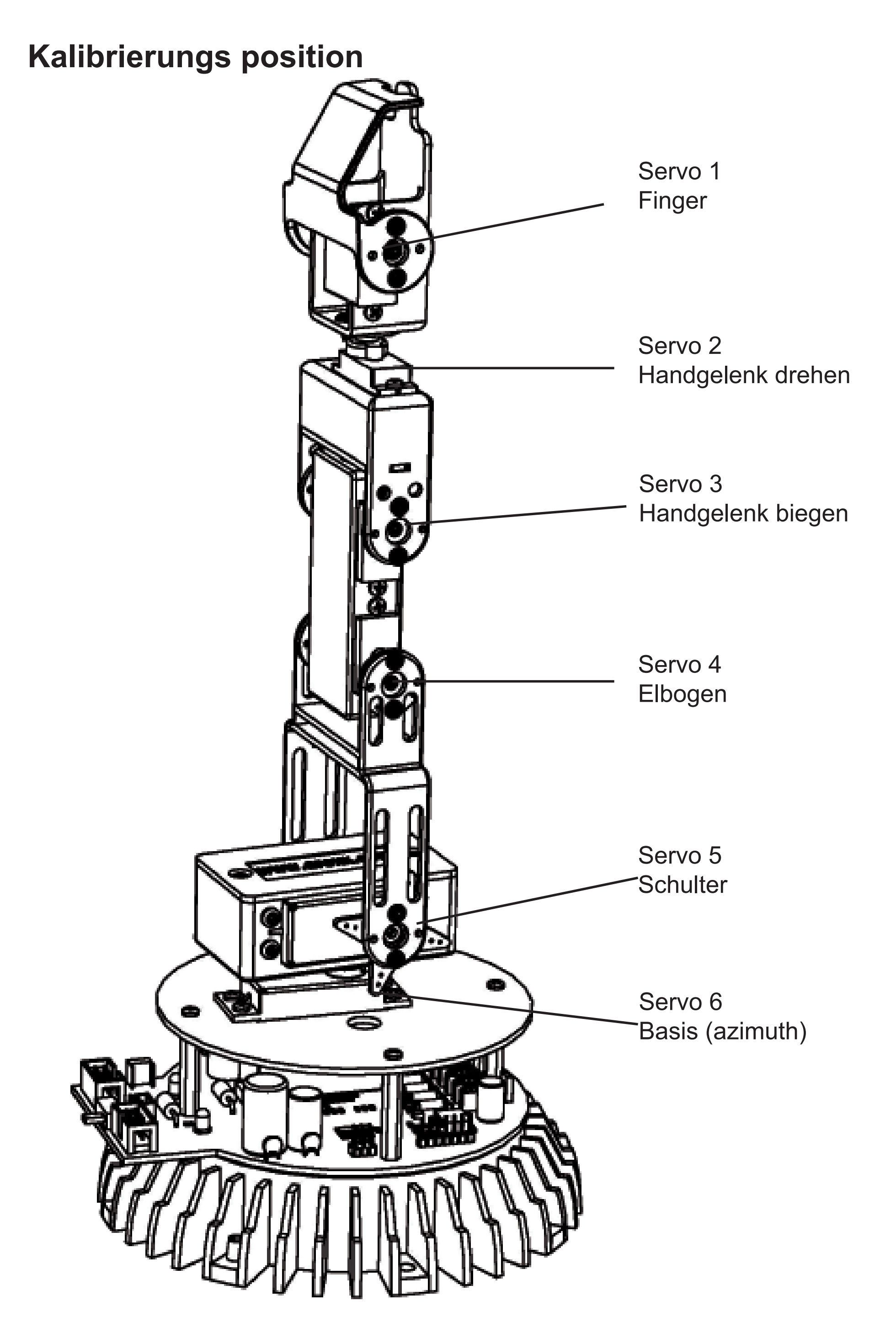




Drehen Sie alle Servomotoren in die Mittellage sodass der Roboter Arm aussieht wie auf Seite 47.
Die Servomotoren 2-6 befinden sich dabei in etwa in der Mittellage und der Finger (Servo 1) ist fast geschlossen.

Wenn die Kalibrierung (C - Calibrate) abgeschlossen ist, können Sie den nachfolgenden Selbsttest des Roboter Arms ausführen. Das Ergebnis der Kalibrierung wird im ATMEGA gespeichert.

Kalibrierungs position



7.8. Tastatur Test

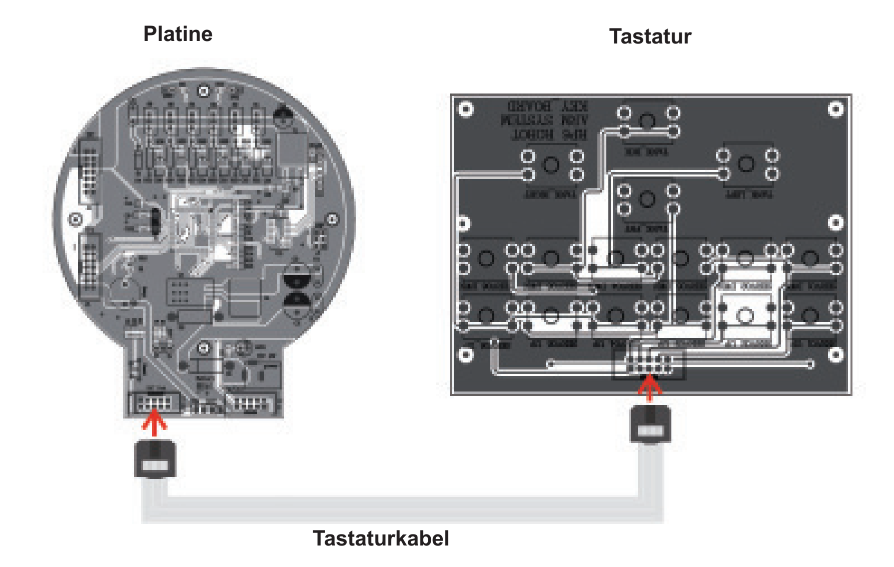
Wir Liefern in diese Bausatz auch eine Tastatur, die auf dem Roboter-Arm angeschlossen werden kann. Es ist eine schöne Option für einfache Vorführungen und erlaubt uns auch die Steuerung eines Roboterarms mit einem Tastenfeld zu üben.

Das Tastenfeld verfügt über 6 Tasten zur Steuerung des Roboter Arms und verfügt über 4 Sondertasten für späteren Erweiterungen.

Zur Überprüfung des Roboter Arms mit der Tastatur müssen wir das benötigte HEX-Programm in den Mikroprozessor des Roboter Arms übertragen.

Wählen Sie dazu die Datei „RobotArm_Key_Board.hex“ in der Liste und aktivieren Sie anschließend den Button „Upload!“ rechts oben unterhalb dem Fortschrittsbalken.

Anschließend können Sie den Roboter Arm einfach mit den Tasten auf der Tastatur bedienen.



8.0. RACS Software

RACS (Robot Arm Control Software) ist die einfachste Methode zur Steuerung und Programmierung des RobotArms. Die Programmierung mit der RACS-Methode verwendet die RobotLoader Software und den USBProgrammier Adapter.

Bevor der Roboter verwendet werden kann muss die HEX Software RAC_PRO.hex in den Flash Speicher des Prozessors geladen werden.

Verbinden Sie das Programmierkabel/Steuerungskabel mit dem USB Port am Computer und starten Sie anschließend die Loader Software. Dabei erscheint folgende Benutzeroberfläche:

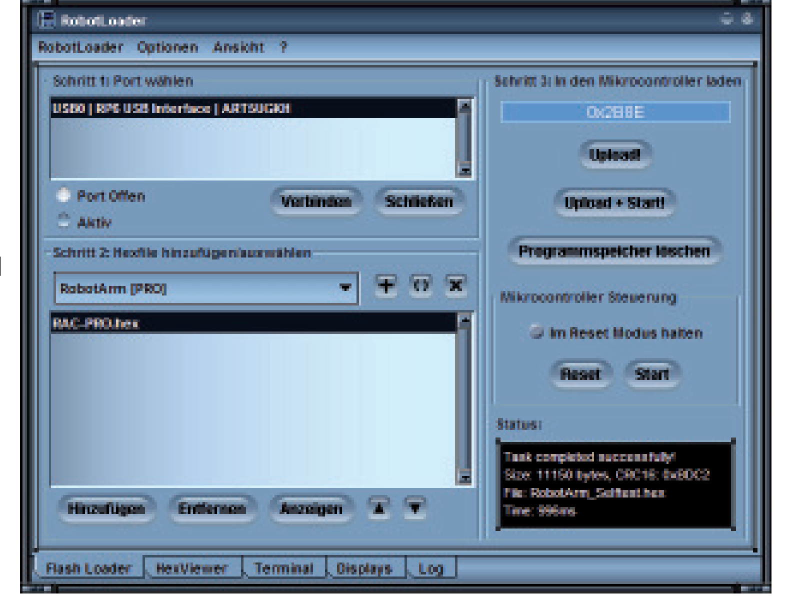


Abbildung 1

Falls kein USB Port in der Liste „Schritt 1: Port wählen erscheint" stellen sie sicher, dass das Kabel verbunden ist und die Treiber des Programmers installiert sind. Sie können die Portliste neu abfragen im Menü;
RobotLoader -> Portliste aktualisieren. Wählen sie anschließen den Port aus und klicken sie auf verbinden.

Im Schritt 2 wählen sie die korrekte .hex Datei aus

– klicken sie dazu auf hinzufügen: **RAC.PRO.HEX**

Anschließend wird die Datei im Schritt 3 mit klick auf die Schaltfläche Upload übertragen.

Für den Betrieb des Roboterarms ist die Verbindung durch klick auf die Schaltfläche schließen im Schritt 1 zu beenden. Falls sie das Programm schließen wird die Verbindung automatisch beendet.

Achten sie darauf, dass keine Verbindung seitens der Loader Software mit dem Roboterarm besteht – der Roboter kann sonst nicht mit der RACS Software gesteuert werden.

7.1. RACS Bedienungsanleitung

Mittels der Software RACS kann der Roboterarm auf einfache Art und Weise gesteuert werden. Es wird eine Verbindung über das Programmier-/Steuerkabel aufgebaut und anschließend können die Motoren des Roboterarms durch Bewegen der Slider mit der Maus verfahren werden. Die aktuellen Positionen können in der Listbox im unteren Teil des Userinterfaces gespeichert, geändert oder gelöscht werden. Somit wird eine Liste mit den einzelnen Positionen erstellt welche als Datei auf dem Computer gesichert werden kann – Schaltfläche save. Ein Laden dieser Stepliste kann zu jedem beliebigen Zeitpunkt erfolgen.



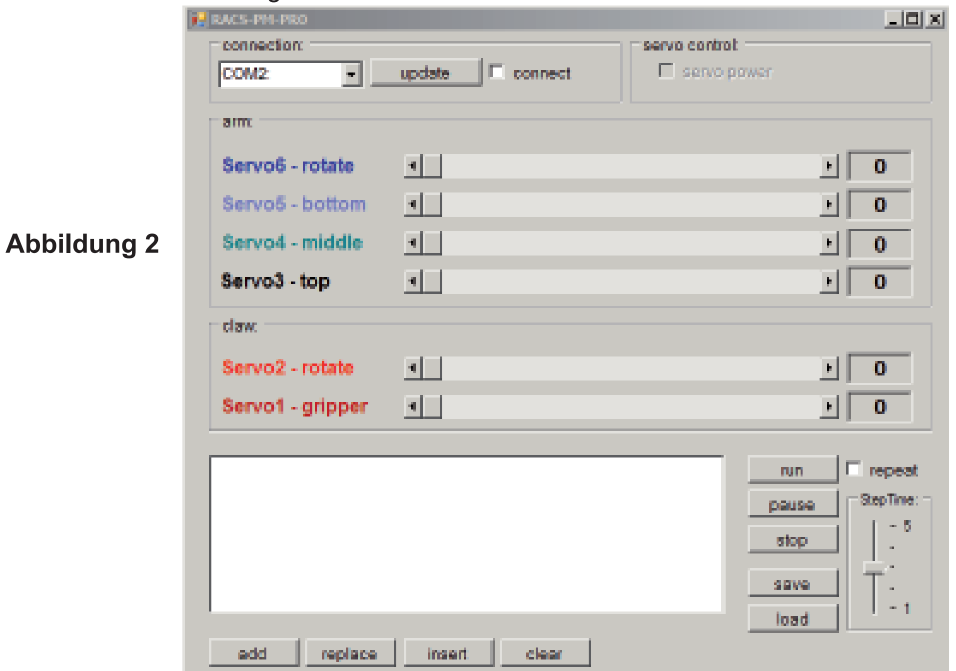
Wichtig !!! Der Roboter überwacht den Motorstrom eines jeden Servomotors. Wird der Grenzwert eines Servos überschritten - z.B.: bei Kollision oder Überlast - beginnt der Text in der RACS Software zu blinken. In diesem Fall muss der Roboter so schnell wie möglich zur letzten Position zurückgefahren oder die Servo Power in der RACS Software deaktiviert werden (deaktivieren der Kontrollbox servo power).

Ansonsten ist eine bleibende Beschädigung des Roboterarmes zu erwarten!!!

7.2. RACS - Verbindungsaufbau

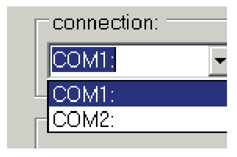
1. Starten der Robot Arm Control Software durch Doppelklick, es erscheint folgende Oberfläche:

Abbildung 2



2. Im Dropdown Menü werden alle seriellen Schnittstellen aufgelistet


Abbildung 3



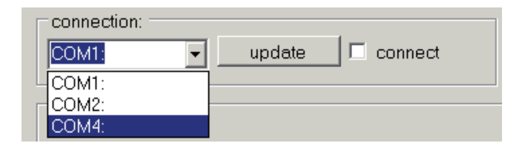
3. Anstecken des USB Programmers

4. Mausklick auf die Schaltfläche Update Bei erneuter Ansicht des Dropdown Menüs wird nun eine zusätzliche Schnittstelle angezeigt. Diese Schnittstelle wurde durch das Anstecken des USB Programmers initialisiert.

Vorsicht: Die Bezeichnung der Schnittstellen wird je nach Rechner unterschiedlich sein!

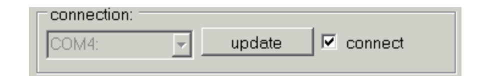
5. diese neue Schnittstelle markieren

Abbildung 4



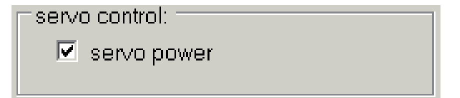
6. Aktivieren der Kontrollbox connect

Abbildung 5



7. Aktivieren der Kontrollbox servo power

Abbildung 6



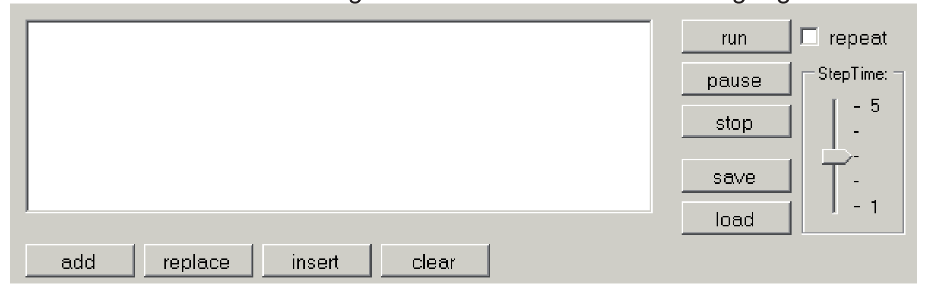
8. Durch Bewegung der Slider können die Servos gesteuert werden. Falls ein Fehler beim Verbindungsaufbau vorliegt wird dies durch folgende Meldung angezeigt. Ein erneuter Verbindungsaufbau ist erforderlich (Schritt 2 bis 6 wiederholen und auf richtige Schnittstelle prüfen).

Abbildung 7



7.3. RACS – automatisierte Positionssteuerung

Im unteren Bereich stehen folgende Steuerelemente zur Verfügung:



Add: mit dieser Schalfläche werden die aktuellen Sliderpositionen in die Liste aufgenommen

Replace: der markierte Listeneintrag wird durch die aktuellen Sliderpositionen ersetzt

Insert: oberhalb des markierten Listeneintrags werden die aktuellen Sliderpositionen eingefügt

Clear: der markierte Listeneintrag wird gelöscht

Save: die Listeneinträge werden in eine Datei gespeichert

Load: die Listeneinträge werden aus einer Datei eingelesen (Vorsicht, die aktuellen Listeneinträge werden gelöscht!)

Run: Die Listeneinträge werden beginnend mit der ersten Position abgearbeitet Ist zusätzlich die Option repeat aktiviert, arbeitet der Roboterarm die Listeneinträge wiederholt ab.

Step Time: Mit Wahl der Step Time wird festgelegt, wie lange der Roboter (in Sekunden) wartet bis die nächste Position in der Liste angefahren wird. Falls in der Liste nur sehr kurze Verfahrwege eingespeichert sind, kann diese Zeit klein gewählt werden. Sind im Gegensatz dazu sehr lange Verfahrwege einprogrammiert – volle 180° Servo Bewegungen – muss diese Zeit größer ge wählt werden, da sonst der Roboter seine Zielposition nicht er reicht und vorzeitig die Position des nächsten Step-Eintrags anfährt.

Pause: Die Abarbeitung wird pausiert

Stop: Die Abarbeitung wird gestoppt

8.0. Programmierung des Robot Arm

Nun kommen wir so langsam zur Programmierung des Roboters.

Einrichten des Quelltexteditors
Erstmal müssen wir uns eine kleine Entwicklungsumgebung einrichten. Der sog. „Quelltext“ (auch Quellcode oder engl. Sourcecode genannt) für unsere C Programme muss ja irgendwie in den Computer eingegeben werden!

Dazu werden wir natürlich auf gar keinen Fall Programme wie OpenOffice oder Word verwenden! Vielleicht ist das nicht für jeden offensichtlich, deshalb wird es hier explizit betont. Damit kann man zwar gut Handbücher wie dieses hier schreiben, aber zum Programmieren ist das absolut ungeeignet. Quelltext ist reiner Text – ohne jegliche Formatierung. Schriftgröße oder Farbe interessieren den Compiler nicht...

Für einen Menschen ist es natürlich übersichtlicher, wenn bestimmte Schlüsselwörter oder Arten von Text automatisch farbig hervorgehoben werden. Das und noch einiges mehr bietet Programmers Notepad 2 (im folgenden kurz „PN2“ genannt), der Quelltexteditor, den wir verwenden werden (ACHTUNG: Unter Linux müssen Sie einen anderen Editor verwenden, der ähnliches wie PN2 bietet. Es sind für gewöhnlich mehrere bereits vorinstalliert! (z.B. kate, gedit, exmacs o.ä.)). Neben dem Hervorheben von Schlüsselwörtern und ähnlichem (sog. „Syntax Highlighting“) gibt es auch eine rudimentäre Projektverwaltung. Man kann so mehrere Quelltextdateien in Projekten organisieren und in einer Liste alle zu einem Projekt gehörenden Dateien anzeigen lassen. Weiterhin kann man aus PN2 komfortabel Programme wie den AVR-GCC aufrufen und so die Programme bequem über einen Menüeintrag übersetzen lassen. Der AVR-GCC ist ja normalerweise ein reines Kommandozeilenprogramm ohne graphische Oberfläche...

Neuere Versionen von Programmers Notepad finden Sie auf der Projekthomepage: ***http://www.pnotepad.org/***

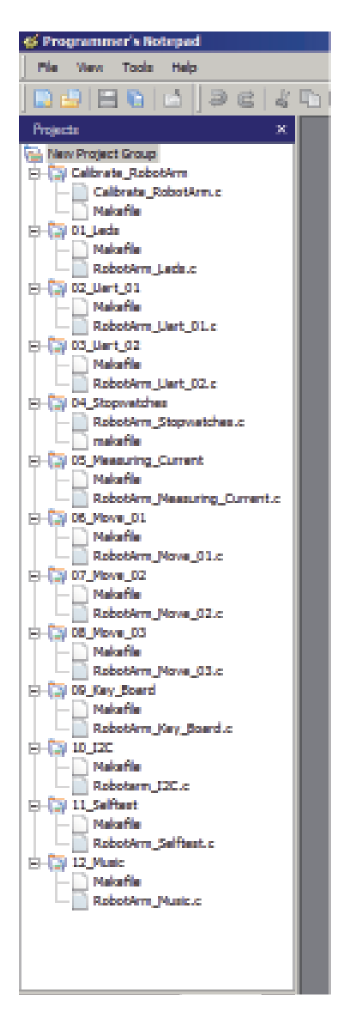
Mit den neuesten Versionen von WINAVR ist es nicht mehr notwendig die Menüeinträge zu erstellen!

ACHTUNG:
In diesem Abschnitt beschreiben wir nicht mehr, wie Sie in PN2 die Menüeinträge erstellen müssen! Mit den neuesten WINAVR Versionen sind diese bereits erstellt.

Siehe Seite 47 „Beispielprojekt öffnen und kompilieren“ wie Sie ein Beispiel Programm öffnen können!

Wenn Sie ein Beispielprojekt geöffnet haben, sollte es im PN2 Schirm etwa so aussehen:

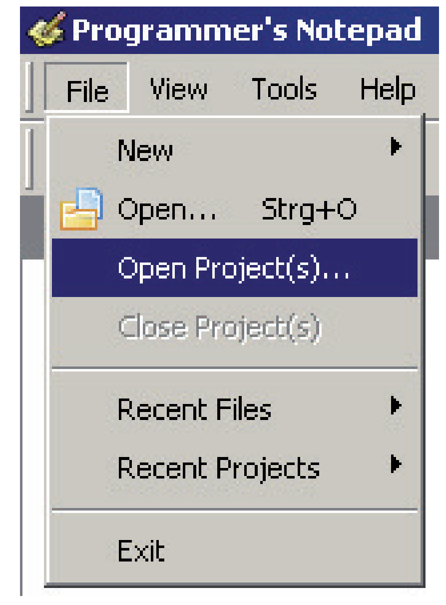
Datei „Robot ArmExamples.ppg“.
Das ist eine Projektgruppe für PN2, die alle Beispielprogramme sowie die Robot Arm Library in die Projektliste („Projects“) lädt.



Links sind alle Beispielprojekte zu sehen, rechts der Quelltexteditor (mit dem angesprochenen Syntax Highlighting) und unten die Ausgabe der Tools (in diesem Fall die Ausgabe des Compilers).

Sie können noch viele andere Sachen in PN2 umstellen und es gibt viele nützliche Funktionen.

Beispielprojekt öffnen und kompilieren



Jetzt probieren wir gleich mal aus, ob auch alles richtig funktioniert und öffnen die Beispielprojekte:

Im Menü „File“ den Menüpunkt „Open Project(s)“ wählen.

Es erscheint ein normaler Dateiauswahl Dialog. Hier suchen Sie bitte den Ordner „Robot Arm_Examples\“ im Ordner in dem Sie die Beispielprogramme gespeichert haben.

Öffnen Sie nun bitte die Datei „Robot ArmExamples [PRO].ppg“. Das ist eine Projektgruppe für PN2, die alle Beispielprogramme sowie die Robot Arm Library in die Projektliste („Projects“) lädt.

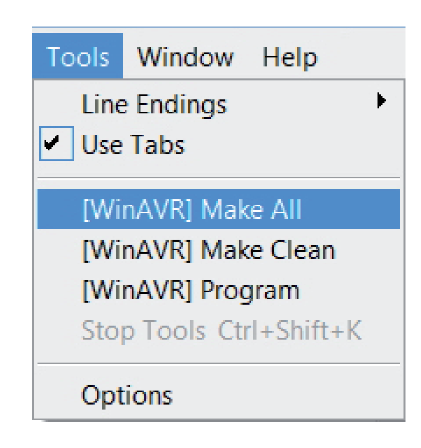
So hat man immer alle Beispielprogramme bequem zur Verfügung und kann sich zu Beginn besser an diesen orientieren oder Funktionen in der Robot Arm Library nachschlagen etc..

Öffnen Sie nun das erste Beispielprogramm ganz oben in der Liste („01_Leds“ und die Datei „01_Leds“ selektieren), die am linken Rand des Programmfensters zu sehen ist! Dazu einfach doppelt auf „01_Leds.c“ klicken! Es erscheint ein Quelltexteditor in einem Fenster innerhalb des Programms.

Unten im Programmfenster von PN2 sollte ein Ausgabebereich zu sehen sein – falls nicht, müssen Sie diesen Bereich im Menü mit View->Output aktivieren ODER, falls er zu klein ist, durch „ziehen“ mit der Maus vergrößern (der Mauscursor verändert sich unten im Programmfenster am oberen Rand des grauen Bereichs in dem „Output“ steht in einem recht schmalen Bereich in einen Doppelpfeil...).

Sie können sich das Programm in dem gerade geöffneten Quelltexteditor schonmal kurz anschauen, allerdings müssen Sie hier noch nicht verstehen, was da genau gemacht wird. Das wird weiter unten noch genauer erklärt. Schonmal vorweg: Bei dem grün eingefärbten Text handelt es sich um Kommentare die nicht zum eigentlichen Programm gehören und nur der Beschreibung/Dokumentation dienen.

Darauf gehen wird später noch genauer ein (es gibt auch eine Version dieses Programms OHNE Kommentare – damit man mal sieht, wie kurz das Programm eigentlich ist. Die Kommentare blähen das schon ziemlich auf, sind aber zur Erklärung notwendig. Die unkommentierte Version ist auch praktisch, um den Code in eigene Programme zu kopieren!).



Zunächst wollen wir nur ausprobieren, ob das Übersetzen von Programmen korrekt funktioniert.

Es sollten oben im Tools Menü die beiden eben angelegten Menüeinträge (s. Abb.) vorhanden sein (oder die standardmäßig in PN vorhandenen [WinAVR] Einträge, das ist egal, es funktioniert normalerweise mit beiden).

Klicken Sie jetzt bitte auf „MAKE ALL"!

PN2 ruft nun die oben angesprochene „make_all.bat" Batch Datei auf. Diese wiederum ruft das Programm „make" auf. Mehr zu „make" folgt noch später.

Das Beispielprogramm wird nun übersetzt (das nennt man „kompilieren" vom englischen „to complile" bzw. „Compiler"=„Übersetzer") und eine Hexdatei erzeugt. Diese enthält das Programm in der für den Mikrocontroller übersetzten Form und kann dann später in diesen geladen und ausgeführt werden! Es werden während der Kompilierung noch viele temporäre Dateien erzeugt (Endungen wie „.o, .lss, .map, .sym, .elf, .dep"). Die brauchen sie alle nicht weiter zu beachten! Mit dem eben angelegten Tool „make clean" kann man diese bequem löschen. Davon ist nur die Hexdatei für Sie interessant! Die Hexdatei wird beim Aufruf von „make clean" übrigens nicht gelöscht.

Es sollte nach dem Aufruf des MAKE ALL Menüpunktes folgende Ausgabe erscheinen (hier jedoch stark gekürzt! Einiges kann natürlich etwas anders aussehen):

```
> "make.exe" all
-------- begin --------
avr-gcc (WinAVR 20100110) 4.3.3
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Size before:
AVR Memory Usage
----------------
Device: atmega64
Program:    3074 bytes (4.7% Full)
(.text + .data + .bootloader)
Data:         68 bytes (1.7% Full)
(.data + .bss + .noinit)
EEPROM:       14 bytes (0.7% Full)
(.eeprom)

Compiling C: Robot Arm_Leds.c
avr-gcc -c -mmcu=atmega64 -I.
-gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -Wall
-Wstrict-prototypes -Wa,-adhlns=./Robot Arm_Leds.lst  -std=gnu99 -MMD -MP -MF
.dep/Robot Arm_Leds.o.d Robot Arm_Leds.c -o Caterpillar_Leds.o

Linking: Robot Arm_Leds.elf
avr-gcc -mmcu=atmega16 -I. -gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-
bitfields
Creating load file for Flash: Robot Arm_Leds.hex
Creating load file for EEPROM: Robot Arm_Leds.eep
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
        --change-section-lma .eeprom=0 --no-change-warnings -O ihex Robot Arm_Leds.elf
Robot Arm_Leds.eep || exit 0
Size after:
AVR Memory Usage
----------------
Device: atmega64
Program:    3074 bytes (4.7% Full)
(.text + .data + .bootloader)
Data:         68 bytes (1.7% Full)
(.data + .bss + .noinit)

EEPROM:       14 bytes (0.7% Full)
(.eeprom)
-------- end -------->
Process Exit Code: 0
> Time Taken: 00:04
```

Wichtig ist ganz unten das „Process Exit Code: 0“. Das bedeutet, dass es beim Übersetzen keinen Fehler gegeben hat. Steht dort ein anderer Code, gibt es einen Fehler im Quellcode, den man korrigieren muss, bevor es klappt. Der Compiler gibt in diesem Fall weiter oben diverse Fehlermeldungen aus, in denen man mehr Infos dazu findet.

Aber bitte beachten Sie, dass „Process Exit Code: 0“ nicht auf ein komplett fehlerfreies Programm hinweist! Denkfehler in Ihrem Programm findet der Compiler natürlich nicht und er kann auch nicht verhindern, dass der Roboter vor die Wand fährt ;-)

WICHTIG: Weiter oben können auch noch Warnungen u.ä. stehen – diese sind oft sehr sehr hilfreich und weisen fast immer auf wichtige Probleme hin! Daher sollten diese immer beseitigt werden! PN2 hebt Warnungen und Fehler farbig hervor, so dass man diese leicht identifizieren kann. Es wird auch die Zeilennummer angegeben, die der Compiler bemängelt. Wenn man auf diese farbig hervorgehobene Meldung klickt, springt PN2 im entsprechenden Editor direkt zu der jew. Zeile.

Auch sehr hilfreich ist die Angabe zum Schluss „AVR Memory Usage“.

Size after:
AVR Memory Usage

```
Size after:
AVR Memory Usage
----------------
Device: atmega64

Program:    3074 bytes (4.7% Full)
(.text + .data + .bootloader)

Data:         68 bytes (1.7% Full)
(.data + .bss + .noinit)
```

Das bedeutet hier, beim Atmega64 Prozessor, dass unser Programm 3074 Bytes groß ist und 68 Bytes RAM für statische Variablen reserviert sind (dazu kommen noch die dynamischen Bereiche für Heap und Stack, das würde an dieser Stelle aber zu weit führen... halten Sie einfach immer mindestens ein paar hundert Bytes Speicher frei). Wir haben insgesamt 64KB (65536 Bytes) an Flash ROM und 2KB (2048 Bytes) an RAM. Von den 64KB sind 2K mit dem Bootloader belegt – also können wir nur 62KB nutzen. Immer darauf achten, dass das Programm auch noch in den verfügbaren Speicher passt!
(Der RobotLoader überträgt das Programm nicht wenn es zu groß ist!)

Bei dem Beispielprogramm oben sind also noch 60414 Bytes frei. Das eigentlich recht kurze Beispielprogramm Example_01_Leds.c ist übrigens nur deshalb schon so groß, weil die Robot ArmBaseLibrary mit eingebunden wird! Also keine Sorge, es ist genug Platz für Ihre Programme vorhanden und so kleine Programme brauchen normalerweise nicht so viel Speicher. Die Funktionsbibliothek benötigt alleine nämlich schon mehrere KB vom Flashspeicher, nimmt Ihnen aber auch sehr viel Arbeit ab und daher werden Ihre eigenen Programme meist relativ klein sein im Vergleich zur Robot ArmBaseLibrary.

Das eben kompilierte Programm kann nun mit dem RobotLoader in den Roboter geladen werden. Dazu fügen Sie die eben erzeugte Hexdatei in die Liste im RobotLoader mit „Add“ bzw. „Hinzufügen“ ein, selektieren diese und klicken auf den „Upload!“ Button, genau wie Sie es auch schon beim Selbsttestprogramm getan haben. Danach können Sie wieder auf den Terminal wechseln und sich die Ausgabe des Programms anschauen. Die Programmausführung muss dazu natürlich zunächst wieder gestartet werden, im Terminal ist es am bequemsten [STRG]+[S] auf der Tastatur zu drücken oder das Menü zu benutzen (oder einfach ein „s“ senden – nach einem Reset müssen Sie allerdings immer etwas warten, bis die Meldung „[READY]“ im Terminal erscheint!). Auch [STRG]+ [Y] ist eine sehr nützliche Tastenkombination, denn damit wird das aktuell selektierte Programm in den Robot Arm geladen und direkt danach gestartet! Man muss also nicht extra vom Terminal aus wieder auf den Karteireiter „Flash Loader“ wechseln oder das Menü benutzen.

Das Beispielprogramm ist sehr einfach und besteht nur aus einem kleinen LED Lauflicht und etwas Textausgabe.

-

Zum Abschluss

Wir hoffen, dass unsere Roboter Ihnen auf den Weg in die Roboterwelt geholfen haben! Wie unsere japanischen Freunde glauben auch wir, dass Roboter nach den Computern und Mobiltelefonen die nächste technologische Revolution bilden werden. Diese Revolution wird auch neue wirtschaftliche Impulse auslösen. Leider haben Japan, andere asiatische Länder und auch die USA, Europa dabei längst überholt. Im Gegensatz zu Europa beginnt der Technikunterricht in Asien bereits in der Grundschule und ist ein wichtiger Bestandteil der Ausbildung.

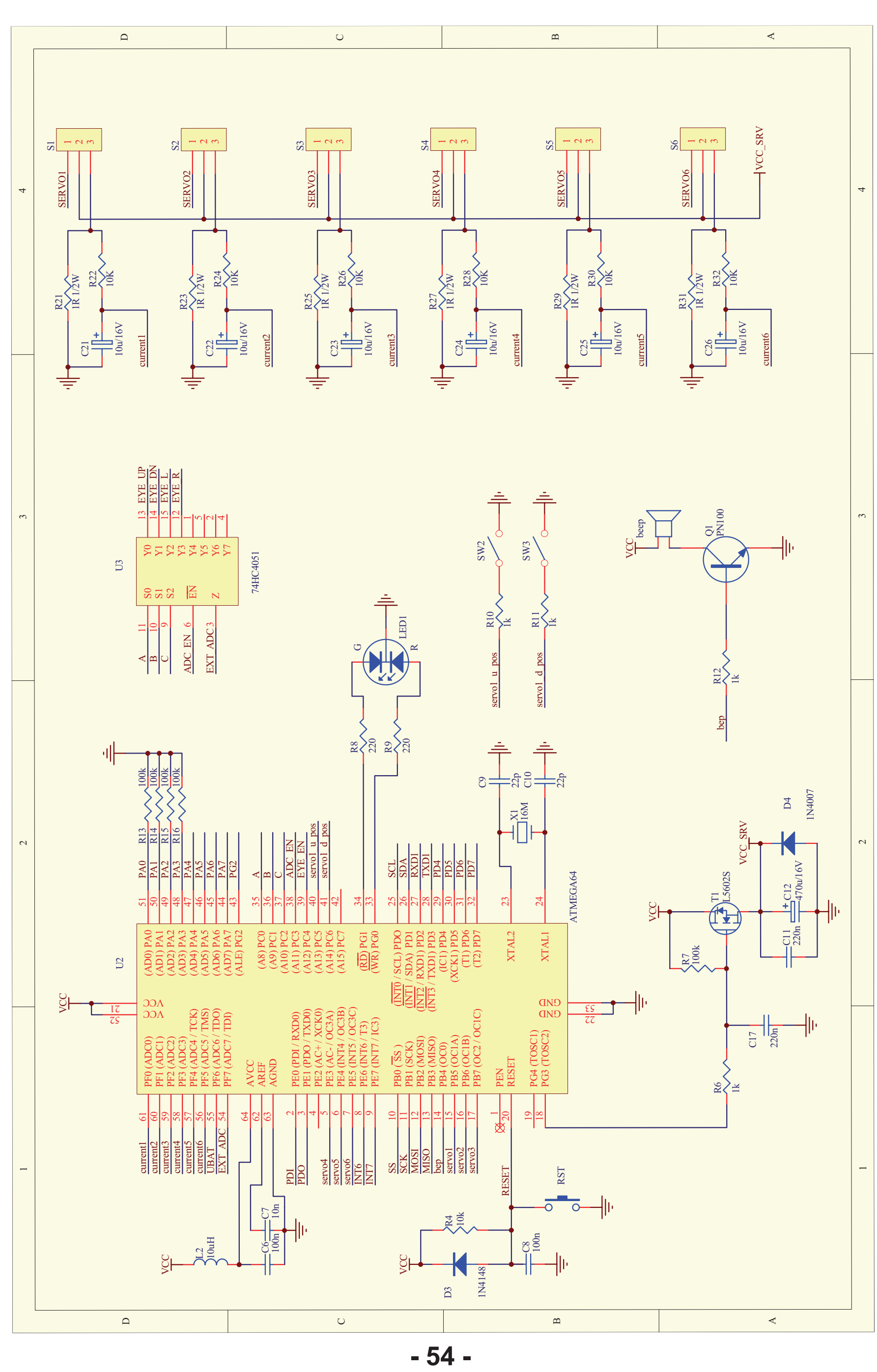
Als Zielsetzung bei der Entwicklung unsere Roboter ASURO, YETI, Caterpillar und Robot Arm haben wir deshalb gewählt:

TO TRAIN A SCIENTIFIC MIND

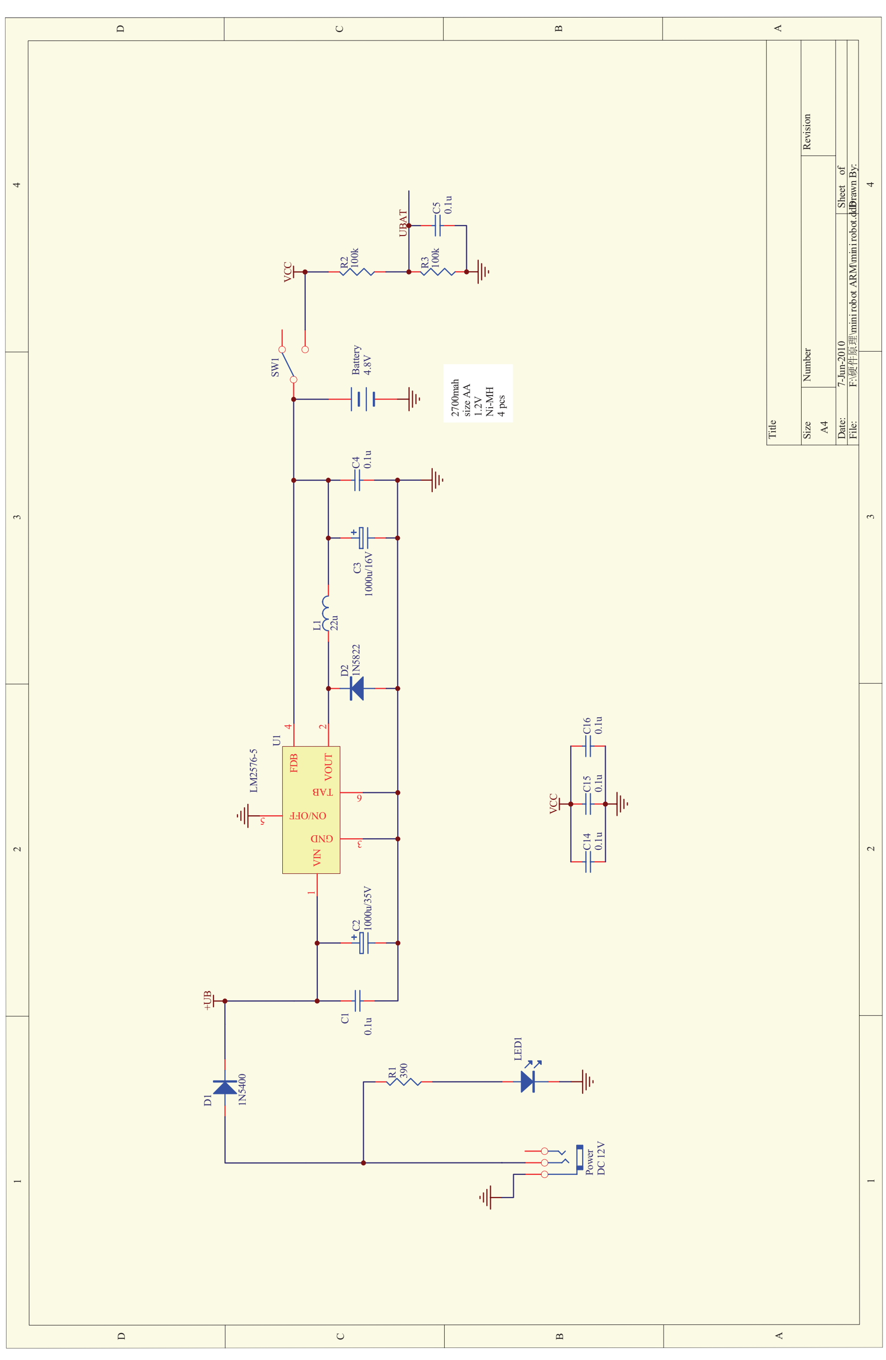


APPENDIX

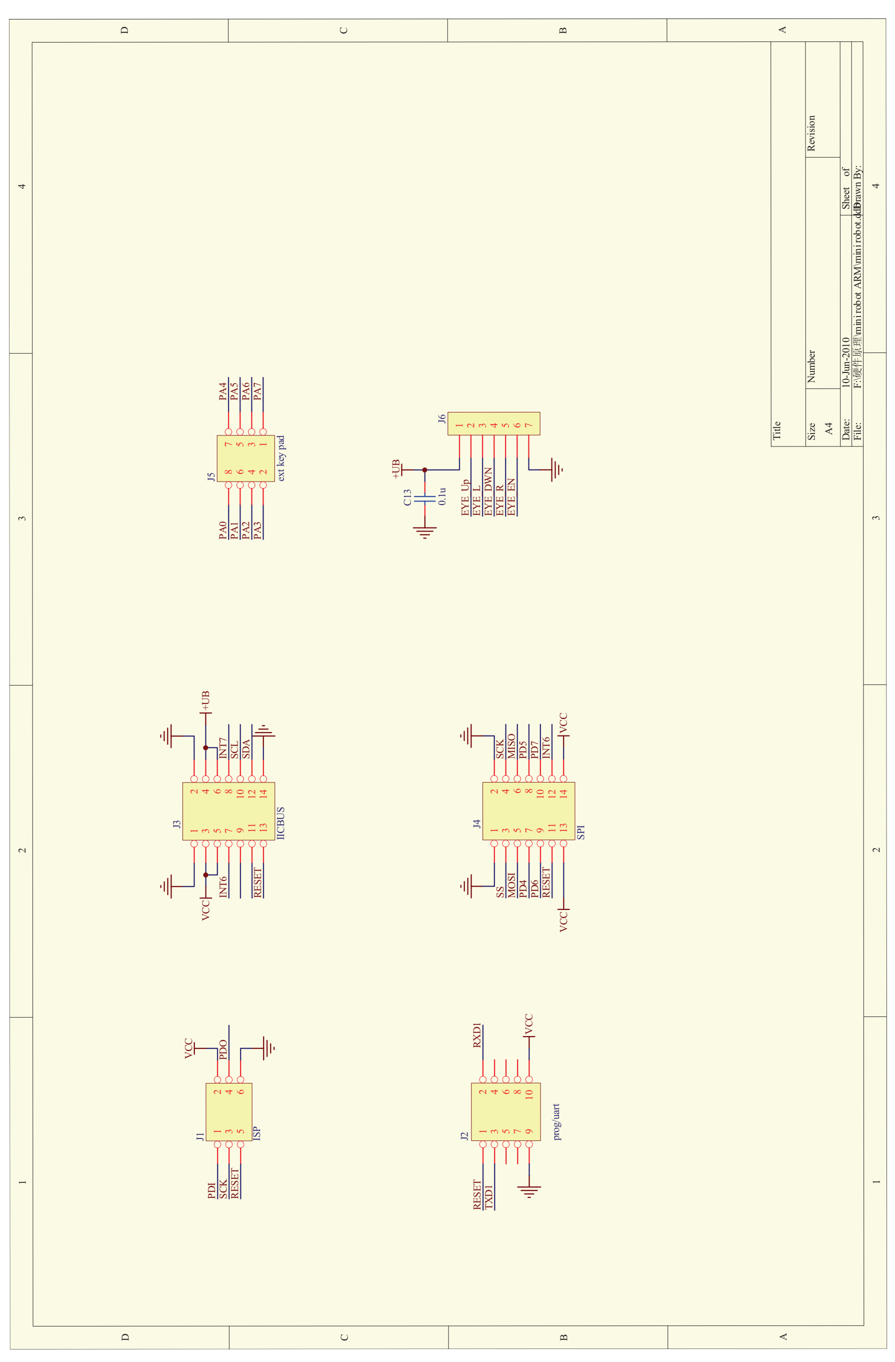
A. SCHALTBILD ROBOT ARM RA1-PRO



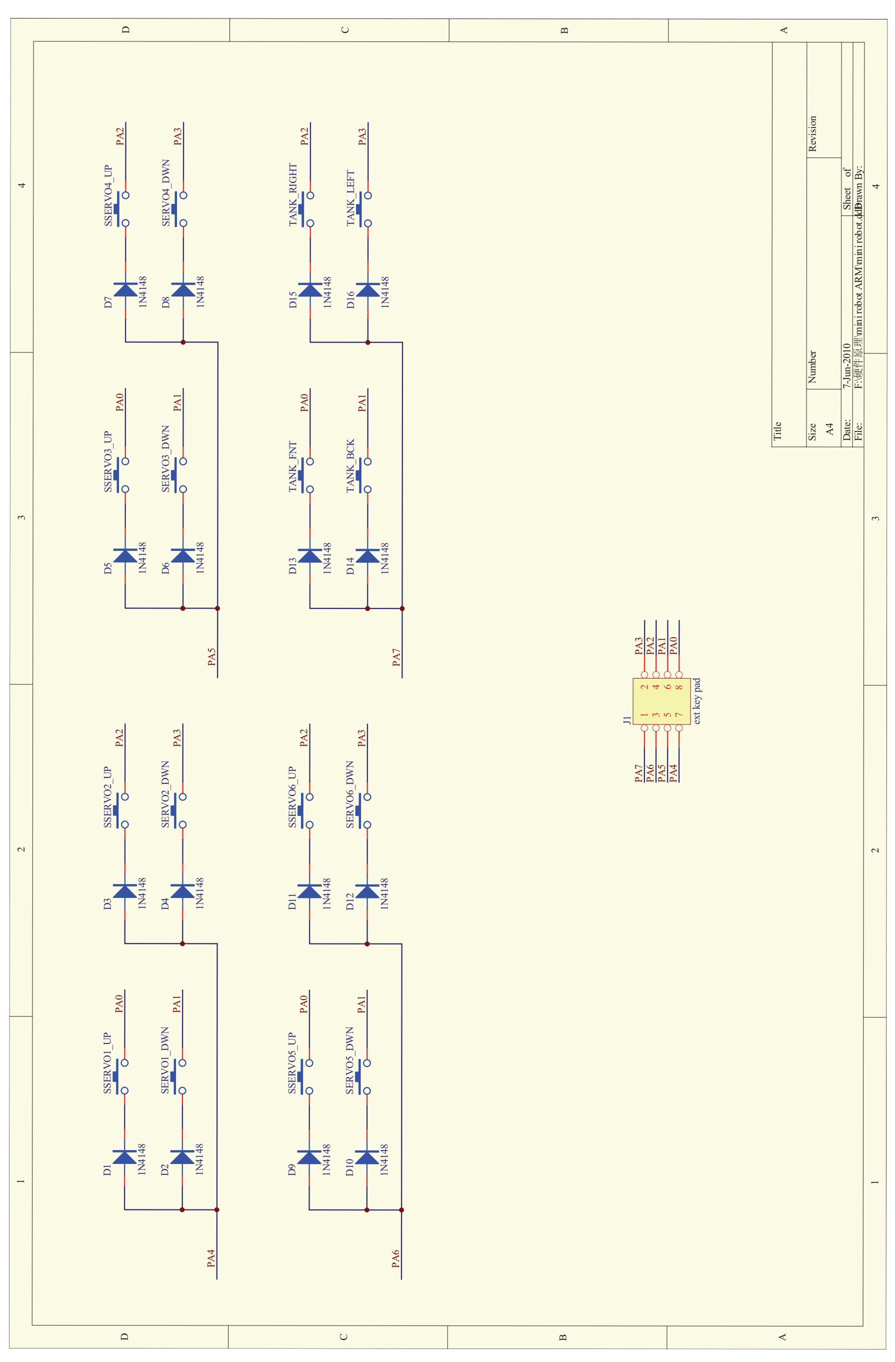
B. SCHALTBILD POWER SUPPLY RA1-PRO



C. SCHALTBILD CONNECTORS RA1-PRO



D. SCHALTBILD TASTATUR RA1-PRO



E. PLATINE ROBOT ARM RA2-HOBBY

